Curb-Tracker: An Integrated Curb Following System for Autonomous Vehicles

Jiahao Liang, Yuanzhe Wang, *Member, IEEE*, Guohao Peng, Zhenyu Wu, and Danwei Wang, *Life Fellow, IEEE*

***Abstract*—Curb following is a critical technology for autonomous road sweeping vehicles. However, existing solutions face two primary challenges: 1) *unreliable curb detection*; and 2) *inefficient motion generation*. *Unreliable curb detection* stems from the wide variability in curb dimensions and types, as well as interference from roadside features, such as vegetation and infrastructure. *Inefficient motion generation* occurs when existing methods prioritize tracking accuracy while neglecting task completion efficiency, leading to prolonged operation times. To address these challenges, we propose Curb-Tracker, an integrated curb-following system designed for autonomous vehicles operating in diverse road environments. First, we develop a robust and adaptive curb detection algorithm that leverages a 2.5-D elevation map of the local environment and dynamically adjusts key parameters online to ensure reliable detection across varying scenarios. Second, to achieve accurate and efficient curb-aligned motion generation, we leverage model predictive contouring control as a tailored framework specifically designed for the curb-following task to generate an optimal control sequence for the vehicle to maintain a specified lateral offset from the curb while maximizing travel progress along it. The proposed system has been implemented on a Hunter 2.0, a front-wheel Ackerman-steering mobile robot, and has been validated through extensive experiments in both Gazebo simulation and real-world environments. Experimental results demonstrate the effectiveness, adaptability, and robustness of the proposed system across a wide range of road scenarios.**

***Index Terms*—Autonomous navigation, autonomous vehicles, trajectory generation.**

I. Introduction

With the rapid advancements in self-driving technology, autonomous vehicles are increasingly deployed in applications, such as last-mile delivery, port operations, and environmental services. Among these, road sweeping stands out as a critical environmental service that requires vehicles to maintain a specified lateral offset while following curbs (as shown in Fig. 1). However, achieving autonomous curb following in real-world scenarios presents significant challenges, which can be broadly categorized into two subproblems: 1) *curb detection*; and 2) *curb-aligned motion generation*. Most current autonomous road sweeping systems rely heavily on manual intervention, where a human operator first drives the vehicle along a designated curb to record a point cloud map and waypoints for subsequent autonomous operation. This approach is not only labor-intensive but also lacks flexibility and adaptability to dynamic and changing road environments. Moreover, during task execution, the vehicle typically operates at a slow and constant speed to ensure tracking accuracy, without adapting to the surrounding road conditions. While this conservative speed control strategy contributes to stability, it often prolongs the overall task duration, result in reducing task completion efficiency. Thus, there is an urgent need for fully autonomous curb-following systems capable of real-time curb detection and curb-aligned motion generation without prior human preparation.

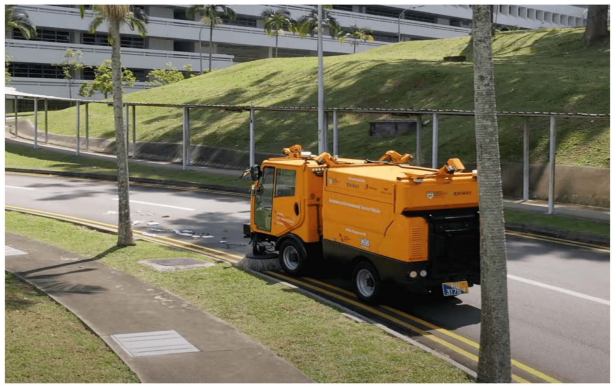


Fig. 1. Illustration of curb following in road sweeping applications. [The autonomous vehicle is required to maintain a specified lateral offset while following the curb to perform road sweeping tasks.]

Curb detection involves identifying and extracting curb information from sensor data to provide accurate curb location and geometry for navigation. The primary challenge lies in the diverse curb dimensions and types, compounded by roadside features, such as vegetation and infrastructure, which increase environmental complexity. While 3-D LiDAR demonstrates good performance for curb detection [1], [2], [3], it is costly and struggles with sparse point cloud density for distant

Received 11 June 2025; revised 15 August 2025; accepted 19 August 2025. Date of publication 12 September 2025; date of current version 26 September 2025. This work was supported in part by the Shandong Provincial Natural Science Foundation, China under Project ZR2025QC1547, and in part by the Key R&D Program of Shandong Province, China under Project 2025CXGC010115. This article was recommended for publication by Associate Editor D. Navarro-Alarcon and Editor R. Murrieta-Cid upon evaluation of the reviewers' comments. *(Corresponding author: Yuanzhe Wang.)*

Jiahao Liang is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, and also with the School of Control Science and Engineering, Shandong University, Jinan 250061, China.

Yuanzhe Wang is with the School of Control Science and Engineering, Shandong University, Jinan 250061, China (e-mail: yuanzhewang@sdu.edu.cn).

Guohao Peng, Zhenyu Wu, and Danwei Wang are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798.

Video of real-world experiments is available at https://https://youtu.be/i42timTckgU.

Digital Object Identifier 10.1109/TRO.2025.3608695

objects, which reduces detection accuracy as distance increases. Cameras, on the other hand, are cost-effective and provide rich texture and color information, making them advantageous for curb detection [4], [5], [6]. However, camera-based methods are sensitive to lighting and weather conditions, and their performance can degrade in challenging environments with occlusions or visual artifacts. Existing methods, while promising under controlled conditions, often rely on manually crafted rules and static assumptions, limiting their robustness in diverse and dynamic environments. Deep learning-based approaches [7], [8], [9] have shown potential to improve generalization and robustness by learning curb features from large datasets. However, they require extensive annotated datasets, which are expensive to create, and their adaptability to unseen scenarios depends heavily on the training data diversity. In addition, the computational resources needed for training and real-time inference pose challenges for deployment on autonomous vehicles with limited processing capabilities.

Curb-aligned motion generation involves generating smooth and accurate motion instructions for the vehicle to follow the detected curb accurately and consistently, despite variations in road conditions and curb geometries. The key challenge lies in accommodating the vehicle's kinematic and dynamic constraints while ensuring safety and task completion efficiency. Existing approaches [10], [11], [12] primarily focus on tracking accuracy but often neglect task completion efficiency, resulting in slow execution speeds. While two-step methods combining trajectory planning and control [13], [14], [15], [16], [17] have been proposed to improve task completion efficiency, the computational cost of generating time-optimal trajectories remains prohibitive for real-time applications.

A closely related research topic to curb following is lane following, a fundamental task in autonomous driving that involves detecting lane markings and controlling the vehicle to maintain its position along the lane's centerline [18], [19], [20], [21]. However, lane following is typically designed for structured road environments with well-defined and consistent lane markings. In contrast, curb following requires handling irregular curb geometries and maintaining precise alignment to avoid collisions. These unique challenges make existing lane-following methods unsuitable for curb-following tasks.

To address the challenges outlined above, this article proposes an integrated curb-following system. First, a robust and adaptive curb detection algorithm based on 2.5-D elevation mapping [22] is developed, which automatically adjusts key parameters online to enhance robustness in diverse scenarios. The proposed model predictive contouring control (MPCC)-based curb-aligned motion generator utilizes these detection results as references, demonstrating precise tracking capability while maintaining high time completion efficiency in real-world implementations. The proposed system has been validated through extensive Gazebo simulations and real-world experiments. The main contributions of this article are as follows.

1) *Robust and Adaptive Curb Detection*: A robust and adaptive curb detection algorithm based on 2.5-D elevation mapping is developed. Unlike traditional methods that rely on manually preset parameters, the developed algorithm dynamically adjusts key parameters in response to environmental changes, ensuring robustness across diverse road conditions with varying curb dimensions and types, as well as interference from roadside features. By leveraging 2.5-D elevation map, the algorithm effectively mitigates the impact of occlusions and noise, enabling robust and accurate curb detection even in complex and dynamic environments.
2) *MPCC-Based Curb-Aligned Motion Generation*: MPCC strategy is introduced for curb-aligned motion generation, which considers both the tracking accuracy and task completion efficiency of curb following. Unlike conventional curb-aligned motion generation methods that prioritize either accuracy or efficiency, the proposed approach integrates both objectives into a unified optimization framework, achieving a dynamic balance that is well-suited for practical applications.
3) *Real-World Implementation*: The proposed system has been implemented on a Hunter 2.0, a front-wheel Ackerman-steering mobile robot. Extensive validation through Gazebo simulations and real-world experiments demonstrates its effectiveness and robustness in varying road conditions.

The rest of this article is organized as follows: Section II reviews related work. A detailed description of the proposed curb-following system is provided in Section III. Section III-B details the proposed curb detection algorithm, while Section III-C introduces the MPCC method for curb-aligned motion generation. Experimental results are presented in Section IV. Finally, Section V concludes this article.

II. Related Work

This section reviews relevant studies on curb detection, curb-aligned motion generation, and lane following, along with recent developments in the primary target application domain of this article, autonomous road sweeping vehicles.

A. Curb Detection

Curb detection focuses on identifying and extracting curb features from sensor data, such as point clouds or images, to provide accurate information about curb locations and geometries for autonomous navigation. Existing approaches for curb detection can be divided into two categories: 1) *handcrafted feature-based methods*; and 2) *deep learning-based methods*.

1) Handcrafted Feature-Based Methods: Handcrafted feature-based methods rely on manually designed rules and algorithms to extract curb features from sensor data by analyzing geometric and spatial characteristics, such as height differences, gradients, and surface normals [2], [3], [23], [24], [25]. One common approach involves using digital elevation maps (DEM) for curb detection. For instance, the authors in [26] and [27] employed DEMs constructed from stereo vision data to identify curbs. In [26], curb feature points were extracted using a Canny edge detector, while random

sample consensus (RANSAC) polynomial fitting is applied to remove outliers. A histogram-based method is then used to locate curb points and determine their heights. Similarly, Kellner et al. [27] computed DEM gradients with a Sobel operator and applied a threshold on the normalized gradient to extract curb feature points, with a maximum search algorithm determining the final curb location and height. However, these methods rely on fixed gradient thresholds, making them less robust in diverse environments where curb gradients can vary significantly due to nearby objects like trees or buildings. Another approach, proposed in [28], estimates plane segments using a mixture of linear regression models on DEMs, defining curbs as the boundaries between these segments. However, this method assumes the consistent presence of curbs between plane segments, which is a limitation in real-world scenarios with diverse curb shapes. In addition to DEM-based techniques, some methods extract curb information directly from raw sensor data without intermediate representations, such as [1], [2], [3], [8], [29], [30]. In [1], [3], curb features are extracted by leveraging the spatial continuity of curb points in point clouds, followed by Gaussian process regression for curb modeling. Similarly, Zhang et al. [2] introduced an improved sliding-beam and search-based method for curb point extraction from point clouds. Jung et al. [8] proposed an optimization framework for curb candidate extraction, incorporating the spatial distribution and smoothness of curb points, and fits the extracted curb points using orthogonal distance regression method. While handcrafted feature-based methods can achieve accurate curb detection under controlled conditions, they often rely on manually tuned parameters. This dependency reduces their robustness in diverse and dynamic environments.

2) Deep Learning-Based Methods: Deep learning-based methods leverage neural networks to automatically learn curb features from large amounts of training data, potentially offering better generalization and robustness compared to handcrafted approaches. In recent years, the rise of deep learning has led to its application in curb detection. For instance, Liang et al. [7] employed a convolutional neural network (CNN) to extract road boundaries from 2D camera images and Lidar point clouds. Reference [31] projects 3D point clouds into 2D bird's-eye view images for guaranteed real-time processing, and then uses a deep CNN to detect curbs. Jung et al. [32] proposed a two-stage deep learning-based method for curb detection that incorporates uncertainty estimation to improve detection accuracy. Gao et al. [9] utilized the STWin Transformer model to extract curb points from voxelized point cloud data, incorporating spatial, channel, and temporal attention mechanisms to improve detection performance. Considering the imbalance in curb point distributions, Zhao et al. [33] introduced a multiscale and channel attention module alongside an adaptive weighted loss function to enhance detection accuracy. Ma et al. [25] proposed an annotation-free curb detection approach that utilizes altitude difference images derived from LiDAR point clouds. Their method incorporates an automatic curb annotation module that generates training data without manual intervention. However, this module relies heavily on heuristics based on human expertise, making the quality of the generated training data dependent on the domain knowledge of experts. FALCON [34], a deep learning-based curb detection model for autonomous driving, fuses RGB and LiDAR data through mirror space attention and dilated spatial pyramid attention module, achieving state-of-the-art performance on the CurbData2024 dataset. Meanwhile, Segment anything in LiDAR (SAL) [35] introduces a text-promptable zero-shot model for LiDAR-based object segmentation and classification. It employs a pseudo-labeling engine that leverages 2-D vision foundation models to provide 3-D supervision, distilling the capabilities of 2-D foundation models into a LiDAR-based SAL model. This paradigm significantly improves performance compared to baselines that lift 2-D features to 3-D space, and it shows strong potential for adaptation to curb detection tasks. Despite their promising results, deep learning-based methods face several limitations. Their performance is highly dependent on the diversity and quality of training data, which can limit generalization to unseen curb types and environments. In addition, these methods typically demand substantial computational resources for both training and real-time inference, posing challenges for deployment on autonomous vehicles with constrained onboard processing capabilities.

In summary, while significant advancements have been made in both handcrafted feature-based and deep learning-based curb detection, much efforts are still needed to provide robust and adaptable curb detection methods for real-world applications.

B. Curb-Aligned Motion Generation

Curb-aligned motion generation involves generating appropriate control inputs to enable a vehicle to follow a detected curb while maintaining a specified lateral offset. Compared to curb detection, curb-aligned motion generation has received relatively less attention. Kim et al. [36] proposed an outdoor navigation system for mobile robots, leveraging curb detection to construct a curb map from 2-D laser scan data. This map is subsequently used for tracking and localization, with a hybrid strategy ensuring safe navigation. However, the system relies on human intervention to correct the robot's motion, limiting its autonomy. Detection-based curb-aligned controllers have also been developed to adjust vehicle motion using lateral tracking errors relative to the curb [10], [11]. For instance, Ali and Mailah [12] presented a navigation framework for three-wheeled mobile robots that employs curb detection results for road following, combining resolved acceleration control and active force control to achieve desired motion. Similarly, Sivakanthan et al. [37] developed a curb recognition and negotiation system for robotic wheelchairs, enabling the detection of curb characteristics and automatic curb negotiation. Wang et al. [38] designed a parameterized alignment control algorithm for curb-aligned motion generation. Experimental results demonstrate that this method achieves high tracking accuracy. However, the vehicle follows the curb at a constant speed, limiting adaptability to complex road conditions and reducing task completion efficiency in dynamic environments. Although these control-based methods demonstrate promising performance, they often lack real-time optimization capabilities to dynamically adjust control inputs for optimal behavior. This limitation makes them less

effective in scenarios that require balancing multiple objectives, such as precise tracking, efficiency, and safety. An alternative approach, inspired by trajectory generation techniques used in autonomous racing or quadrotor racing, divides the curb-aligned motion generation into two subproblems: 1) planning, which computes a time-optimal and dynamically feasible trajectory based on curb detection results; and 2) control, which focuses on accurately tracking the planned trajectory. However, these optimization-based methods are often challenged by the computational demands of time allocation, making them less suitable for real-time curb-aligned motion generation applications [13], [14], [15], [16], [17].

Consequently, developing advanced curb-aligned motion generation methods that integrate real-time optimization is essential for enhancing performance in complex, real-world applications.

C. Lane Following

Lane following, a closely related research topic, has been extensively studied in the field of autonomous driving [18], [19], [20], [21]. This task involves detecting lane markings and controlling the vehicle to maintain its position along the lane centerline. For instance, the authors in [18] and [20] introduced automated lane-keeping systems that employ vision-based lane detection algorithms to identify lane markings and lateral control algorithms to keep the vehicle within lane boundaries. These systems are primarily designed for structured road environments with well-defined and consistent lane markings. Notably, traditional lane-following methods typically focus exclusively on lateral control, often neglecting integration with longitudinal control. While this focus may suffice for lane following, it poses limitations for curb following, where changes in vehicle speed can significantly impact the performance of lateral controllers. Such speed variations, though tolerable in lane following, can lead to safety risks in curb following, where precise control is critical to avoid collisions with curbs. In recent years, deep learning-based approaches for lane following have also gained traction [21], [39], [40]. These methods process image data to predict vehicle steering angle and speed, enabling end-to-end control. However, they are not directly applicable to curb-following tasks. Unlike lane following, which focuses on maintaining a position along the lane centerline, curb following requires the vehicle to maintain a specific lateral distance from the curb, which may vary depending on road conditions or task-specific requirements.

In summary, while lane-following methods have been widely explored and successfully applied in autonomous driving, they are not directly transferable to curb-following tasks due to the distinct challenges involved. Developing specialized curb-following methods that address these challenges is essential for ensuring safe and efficient navigation in autonomous road sweeping vehicles.

D. Autonomous Road Sweeping Vehicles

In recent years, a number of companies have introduced commercial autonomous sweeping vehicles aimed at automating urban cleaning tasks. For instance, Hako [41] offers the Citymaster series, which features assisted driving capabilities for road-cleaning operations. Other manufacturers, including Autowise.ai [42], Gaussian Robotics [43], and Enway [44], have developed autonomous sweepers designed for outdoor urban environments, such as streets, campuses, and municipal areas. Enway's V20e and B2 models, built on Bucher Municipal vehicle platforms, operate fully autonomously by leveraging high-precision curb-aligned navigation, vision and LiDAR fusion, remote operator services, and autonomous charging. Notably, Enway's compact Donner series road sweeper has received approval from Singapore's Land Transport Authority for use on public roads, demonstrating successful deployment in dynamic traffic environments. In the academic domains, an autonomous robotic road sweeper is developed based on the commercial KMR 1200 platform (Alfred Kärcher GmbH, Germany), incorporating mechanical, electronic, and navigation system upgrades to enable automated cleaning in structured environments with both static and dynamic obstacles [45]. In another work, a commercial manual brush sweeper is retrofitted with dual fish-eye cameras to enable low-cost curbside cleaning. The system utilizes curb recognition and tracking from top-view images to guide the sweeping process [46]. Despite these developments, achieving robust adaptability in unseen road environments remains a significant challenge, particularly in online curb detection and accurate curb-aligned motion generation. These limitations motivate the method proposed in this article, which aims to address the robustness and adaptability gaps in current autonomous sweeping systems.

III. Methodology

A. Preliminaries

1) Notations: In this article, we define two primary coordinate frames:

- *World frame* W: The world frame is a fixed reference frame in the environment, with its origin defined as the origin of the point cloud map.
- *Vehicle frame* B: The vehicle frame is attached to the vehicle, where the x-axis aligns with the forward-moving direction of the vehicle.

We adopt the notation convention where the pre-superscript indicates the reference frame where a variable is expressed, and the post-subscript denotes the time step k when a variable is evaluated. For example, the vehicle position with respect to W at time step k is represented as ${}^{W}\boldsymbol{p}_k$. For simplicity and without loss of clarity, variables without a presuperscript are assumed to be expressed with respect to the world frame W.

2) Vehicle Model: In this article, we consider a front-wheel Ackerman-steering vehicle described by the following kinematic model:

$$\begin{aligned}\dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \frac{v}{L}\tan\delta \end{aligned} \tag{1}$$

where $\boldsymbol{p}_v = [x\ y]^\top$ represents the position of the rear-wheel center, θ denotes the orientation of the vehicle in the world frame,

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

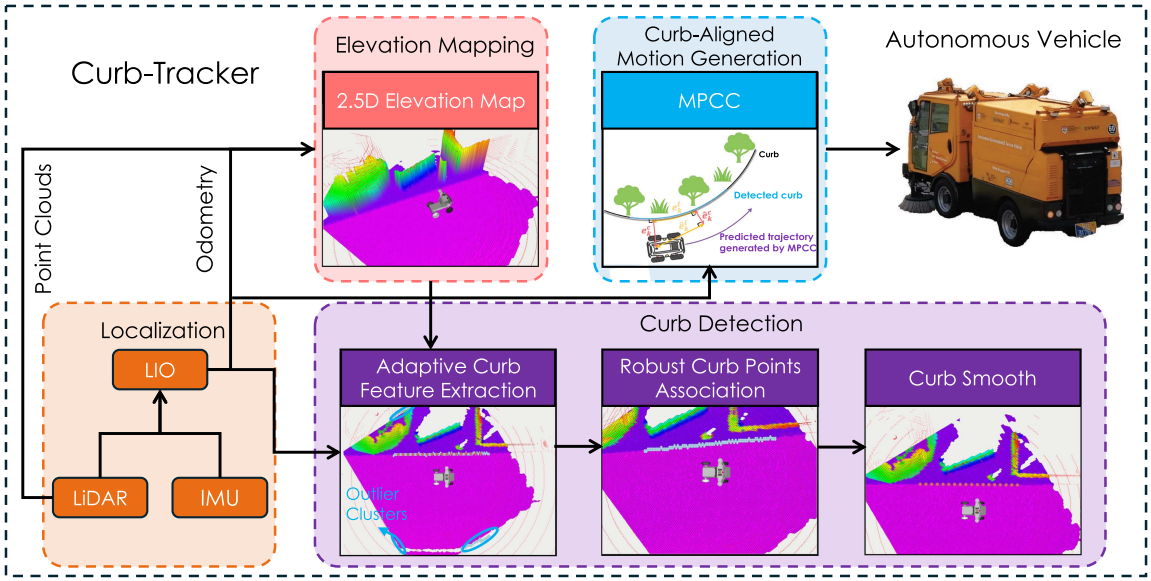


Fig. 2. Schematic diagram of the proposed curb-following system. [The system consists of two main components: a robust and adaptive curb detection module and a MPCC-based curb-aligned motion generation module. The perception component extracts accurate and reliable curb information, which is subsequently utilized by the control component to generate optimal control commands, enabling the vehicle to follow the curb effectively. In addition, an LiDAR inertial odometry (LIO)-based localization component provides accurate localization results for curb detection, curb-aligned motion generation, and elevation mapping component, respectively.]

v represents the velocity of the vehicle, L is the wheelbase, and δ is the steering angle. The vehicle state is defined as $\boldsymbol{x} = [x\ y\ \theta]^\top$, while the control input is represented by $\boldsymbol{u} = [\delta\ v]^\top$.

3) Curb Detection: The curb detection problem can be divided into two subproblems as follows.

- *Curb localization*: This involves identifying the curb within the point cloud data and generating a series of points along the curb, denoted as $\mathbf{P}_{\text{cur}} = [{}^B\mathbf{p}_1, {}^B\mathbf{p}_2, \ldots, {}^B\mathbf{p}_N]$.
- *Curb parameterization*: This focuses on representing the curb with a mathematical model. In this article, we adopt a Bézier curve for curb parameterization. A Bézier curve of order n is defined as

$$\boldsymbol{p}(t) = \sum_{i=0}^{n} \mathbf{C}_i b_i^n(t) \tag{2}$$

where $b_i^n(t) = C_n^i t^i (1-t)^{n-i}$ is the ith Bernstein polynomial of degree n, C_n^i is the binomial coefficient, and $\mathbf{c}_i$ represents the ith control point. In this article, we use a cubic Bézier curve ($n = 3$) for both simulation and real-world experiments. This choice is sufficient to model curbs in most practical scenarios.

4) Model Predictive Contouring Control: The primary objective of MPCC method is to minimize contour errors and maximize travel progress along a predefined geometric path [47]. The predefined path is parameterized by arc length s. Consequently, the cost function of MPCC is formulated as follows:

$$J_{\text{MPCC}}(\boldsymbol{x}) = \sum_{k=1}^{N} w_c \, ||e_k^c - l|| - \rho s_N \tag{3}$$

where N is the prediction horizon, $e_k^c = ||\boldsymbol{p}_{v_k} - \boldsymbol{p}(s_k)||$ represents the contour error, and $\boldsymbol{p}(s_k)$ is the projection of the vehicle's position $\boldsymbol{p}_{v_k}$ onto the curb curve. The term l denotes the desired lateral offset from the curb, w_c is the weight assigned to the contour error, and ρ is the weight for travel progress, designed to maximize movement along the curb. Lastly, s_N represents the total travel progress along the predefined geometric path at the end of the prediction horizon. By minimizing (3), a sequence of optimal control commands $\boldsymbol{u}^*$ is generated. Further details on how to apply MPCC to curb-aligned motion generation are described in Section III-C.

5) System Overview: The overview of the proposed curb-following framework is shown in Fig. 2. Taking LiDAR point clouds as input, a vehicle-centric elevation map is constructed to capture detailed environmental information around the vehicle. To robustly and accurately extract curb feature points, an adaptive curb feature extraction method is applied to the elevation map. This method processes the elevation map by dynamically adjusting key parameters based on changing road conditions, eliminating the need for any predefined threshold parameters commonly used in traditional handcrafted-based approaches. Subsequently, a curb association method matches the extracted curb feature points with historical data, enabling the removal of outliers. The matched curb points are then represented as a cubic Bézier curve. To further enhance the accuracy of curb detection and the performance of curb-aligned motion generation, a Kalman filter is employed to fuse the current detection results with historical data. Finally, the fused curb points serve as a reference for the MPCC-based curb-aligned motion generation method, which generates optimal control commands for the vehicle to follow the curb.

B. Robust and Adaptive Curb Detection

Identifying curb points in sensor data is a challenging task, especially considering the variability in curb dimensions, types,

and the presence of roadside features, such as vegetation and infrastructure. As discussed in Section II-A, the performance of existing curb detection methods based on manually designed rules is highly dependent on predetermined parameters, such as height and gradient thresholds, which are typically derived from human experience, like the methods in [3], [26], and [27]. As a result, these methods lack robustness and frequently underperform in diverse environments. To overcome these limitations and reduce reliance on manually defined parameters, we propose a robust and adaptive curb detection algorithm based on a 2.5-D elevation map. This method enhances generalization and robustness by incorporating adaptive techniques that accommodate varying road conditions.

1) Adaptive Curb Feature Extraction: A *curb* is typically defined as the boundary of the drivable area on a road. For normal vehicles, objects with a height approximately 0.3 to 0.5 times the wheel diameter $\mathcal{D}_w$ are generally considered as impassable obstacles. Consequently, the elevation difference between ground and nonground regions serves as a key indicator for detecting road boundaries. To extract curb features from the elevation map, it is essential to distinguish ground and nonground grid cells. Once this classification is completed, curb features can be identified by detecting the boundary interface between ground and nonground grid cells. To achieve this, we first define a fixed search region $\mathcal{R}$ around the grid cell corresponding to the robot's position to identify potential curb grid indexes. Within this region, a rowwise detection is performed in the direction of the expected curb. The first grid cell in each row with an elevation difference exceeding $0.3D_{\text{wheel}}$ is identified as a potential curb point. If no such grid cell is found within the defined search region for a particular row, the boundary grid cell in that row is considered the edge of the search area and is included as a potential curb point. The description of the process is shown in Fig. 3. After determining the boundary, all grid cells within it are classified as ground grid cells, denoted as $\mathcal{G}$. Based on $\mathcal{G}$, the ground height $\bar{h}$ and deviation σ for the current road scenario are estimated by

$$\bar{h} = \frac{1}{N_{\mathcal{G}}} \sum_{i,j \in \mathcal{G}} \mathcal{M}(i,j)$$
$$\sigma = \sqrt{\frac{1}{N_{\mathcal{G}}} \sum_{i,j \in \mathcal{G}} (\mathcal{M}(i,j) - \bar{h})^2} \tag{4}$$

Using these online estimation values, a binary Gaussian kernel is designed and convolved with the elevation map to differentiate ground grid cells from nonground grid cells. The convolution process yields a binary elevation map, denoted as $\mathcal{M}^*$

$$\mathcal{M}^*(i,j) = \begin{cases} \bar{h}, & \text{if } \bar{h} - 3\sigma < \mathcal{M}(i,j) < \bar{h} + 3\sigma \\ 0.1\bar{h}, & \text{otherwise} \end{cases} \tag{5}$$

where $\mathcal{M}(i,j)$ represents the elevation value of the grid cell at position (i,j) in the elevation map. Subsequently, we use Canny operator to process $\mathcal{M}^*$ to extract curb feature points, denoted as $\mathbf{P}_{\text{ftr}}$.

By dynamically adjusting the thresholds according to environmental changes, the proposed method can effectively extract curb feature points from the elevation map, ensuring accurate and reliable curb detection, and reducing the reliance on manually defined parameters.

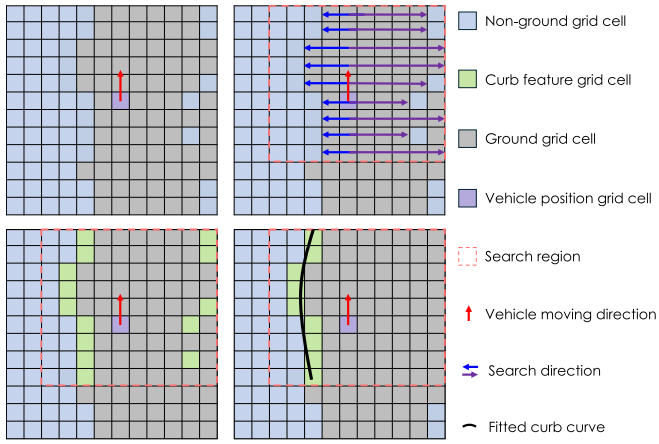


Fig. 3. Illustration of the proposed robust and adaptive curb detection algorithm. [**Top Left**: The elevation map is constructed from the LiDAR point clouds to capture detailed environmental information around the vehicle. Top Right: A search region $\mathcal{R}$ is defined around the grid cell corresponding to the robot's position. Within the search region, a rowwise detection is performed to identify ground grid cells. Then, the ground grid cells are used to estimate the ground height $\bar{h}$ and deviation σ for the current road scenario. Bottom Left: A binary Gaussian kernel is designed and convolved with the elevation map to extract curb feature points within the search region. **Bottom Right**: After robust curb points association, the outliers are filtered out and the curb points are fitted to a cubic Bézier curve.]

2) Robust Curb Points Association: Sensor noise and odometry drift can introduce significant errors in the elevation map, causing some ground grid cells to be misclassified as curb features. In addition, the sparsity of LiDAR point clouds at greater distances may lead to unstable height values near the edges of the elevation map, resulting in potential outliers. These factors not only compromise the accuracy of curb detection but also elevate the risk of false positives. To address these challenges, further processing is required to filter out noise and outliers, ensuring that only the most reliable curb points are identified for subsequent use. To robustly identify the curb points, we first apply DBSCAN [48] to cluster the curb feature points $\mathbf{P}_{\text{ftr}}$ in the current frame, producing a set of clusters $\{\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_n\}$. To determine the curb cluster, we compare the similarity between these clusters and the historical fused curb points set $\mathbf{H}_f$, which will be introduced in Section III-B3. If $\mathbf{H}_f$ is empty, the cluster containing the most points is selected as the curb cluster. Otherwise, Chamfer distance is used to determine the curb cluster $\mathbf{O}_{\text{curb}}$. The Chamfer distance is defined as

$$d_{\text{CD}_i}(\mathbf{O}_i, \mathbf{H}_f) = \frac{1}{N_{\mathbf{O}_i}} \sum_{\mathbf{p}_{\text{ftr}} \in \mathbf{O}_i} \min_{\mathbf{p}_f \in \mathbf{H}_f} \|\mathbf{p}_{\text{ftr}} - \mathbf{p}_f\|_2^2 + \frac{1}{N_{\mathbf{H}_f}} \sum_{\mathbf{p}_f \in \mathbf{H}_f} \min_{\mathbf{p}_{\text{ftr}} \in \mathbf{O}_i} \|\mathbf{p}_f - \mathbf{p}_{\text{ftr}}\|_2^2 \tag{6}$$

where $N_{\mathbf{O}_i}$ and $N_{\mathbf{H}_f}$ are the number of points in the point sets $\mathbf{O}_i$ and $\mathbf{H}_f$, respectively, and $\mathbf{p}_{\text{ftr}}$ and $\mathbf{p}_f$ are points within these sets. The cluster with the smallest Chamfer distance to $\mathbf{H}_f$ is then

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

selected as the curb cluster, denoted $\mathbf{O}_{\text{curb}}$, ensuring consistency with the previously detected curb points.

3) Curb Smoothing: The detection results from the current frame cannot be directly utilized for curb-aligned motion generation because the elevation map is highly susceptible to sensor noise and odometry drifts. This sensitivity may cause abrupt changes between consecutive frame detections, leading to inconsistent and inaccurate curb following. To address this issue, we fuse the current frame detection results with historical data to ensure accurate and consistent curb detection over time. First, the current frame curb cluster is fitted to a cubic Bézier curve. Then, a series of curb points are sampled from the fitted curve. For each sampled point, it is fused with the best-matched point in the historical fused curb points set $\mathbf{H}_f$ using Kalman filter. The details of curb fusion are described as follows.

a) Curb parameterization: In this article, a cubic Bézier curve is used to model the geometric shape of the curb. The process begins with reparameterizing the curb cluster using the chord length parameterization method. Next, the cubic Bézier curve is fitted to the data points using the least squares method. Finally, the curb curve represented by the B'ezier curve is obtained.

b) Curb fusion: Detection results from a single frame cannot be directly used for curb-aligned motion generation due to the susceptibility of the elevation map to sensor noise and odometry drift. This factor can lead to inaccuracies in curb detection, making the results unreliable for guiding curb-following tasks and increasing the risk of deviations from the intended path. To overcome these challenges, we use Kalman filter [49] to fuse the current frame detection results with historical data to ensure accurate and consistent curb detection over time.

By minimizing reliance on manually defined parameters, this approach significantly enhances robustness and adaptability, making it effective across a wide range of scenarios.

C. MPCC-Based Curb-Aligned Motion Generation

Curb following requires the vehicle to move precisely along the detected curb while maintaining a specified lateral offset and ensuring high task completion efficiency. In this article, we address this challenge using the MPCC approach, which simultaneously incorporates these objectives and enables real-time implementation. The proposed MPCC-based curb-aligned motion generation method is detailed in the following.

1) Mathematical Modeling:

a) Error definition: Given the fused point set $\mathbf{H}_f$, the cubic spline method is used to fit the $x-y$ coordinates of the fused curb points, resulting in an arc-length parameterized 2-D curve denoted as ${}^{c}\boldsymbol{p}(s)$. The objective of curb-aligned motion generation is to minimize the deviation between the normal distance from the curb curve ${}^{c}\boldsymbol{p}(s)$ and the desired offset l, while maximizing the travel progress s along the curve. The contour error e_k^c at time step k is thus defined as

$$e_k^c = \min_s \left\| \boldsymbol{p}_{v_k} - {}^{c}\boldsymbol{p}(s) \right\|_2 . \tag{7}$$

However, incorporating this term directly into (3) is computationally expensive due to its inner optimization structure. To address this, we approximate the optimal solution of (7) with $\hat{s}$, determined as

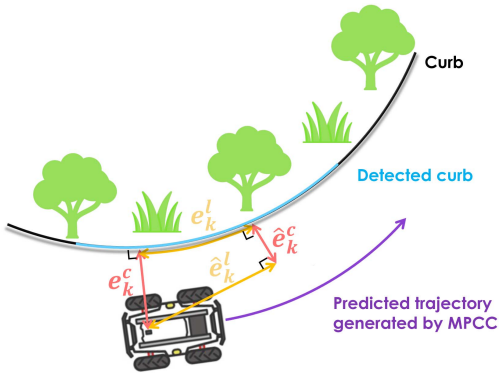


Fig. 4. Illustrative diagram of contour error e_k^c and lag error e_k^l, along with their respective approximations $\hat{e}_k^c$ and $\hat{e}_k^l$. [The travel progress of the projection point corresponding to vehicle's current position is denoted as s_k^*, while $\hat{s}_k$ represents its approximation. A smaller $\hat{e}_k^l$ indicates more accurate approximations.].

$$\hat{s}_k = \hat{s}_{k-1} + v_{s_k} \Delta T \tag{8}$$

where v_{s_k} represents the speed of a virtual target point moving along the curb curve, and ΔT is the sampling time period. Based on this approximation, a new error term, referred to as $lag\ error$, is introduced into the cost function of MPCC-based curb-aligned motion generation

$$e_k^l = \left\| \hat{s}_k - s_k^* \right\|_2 . \tag{9}$$

The *lag error* evaluates the quality of the approximation of the optimal solution to the contour error. To simplify the optimization problem, the approximations of the contour error and lag error are defined as follows:

$$\begin{aligned} \hat{e}_k^c\left(\boldsymbol{p}_k, s_k\right) = & \sin\theta\left(s_k\right)\left(x_k - x_d\left(s_k\right)\right) \\ & - \cos\theta\left(s_k\right)\left(y_k - y_d\left(s_k\right)\right) \\ \hat{e}_k^l\left(\boldsymbol{p}_k, s_k\right) = & -\cos\theta\left(s_k\right)\left(x_k - x_d\left(s_k\right)\right) \\ & - \sin\theta\left(s_k\right)\left(y_k - y_d\left(s_k\right)\right) \end{aligned} \tag{10}$$

where $\boldsymbol{p}_{v_k} = [x_k\ y_k]^\top$ denotes the position of the vehicle at the time step k, and $\theta_k = arc\tan(\frac{y(s_k)}{x(s_k)})$ is the tangent angle of the curve of the curb at the approximated point $\boldsymbol{p}(s_k) = [x_d(s_k)\ y_d(s_k)]^\top$. The *contour error*, *lag error*, and their approximations, $\hat{e}_k^c$ and $\hat{e}_k^l$, are depicted in Fig. 4 .

b) Augmented system model: To enhance the smoothness of the curb-aligned motion generation process and mitigate vehicle oscillations, the rate of change of control inputs is incorporated into the system dynamics. Building upon previous works [50], [51], an augmented system model is introduced

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \frac{v}{L}\tan\delta \\ \dot{s} &= v_s \end{aligned}$$

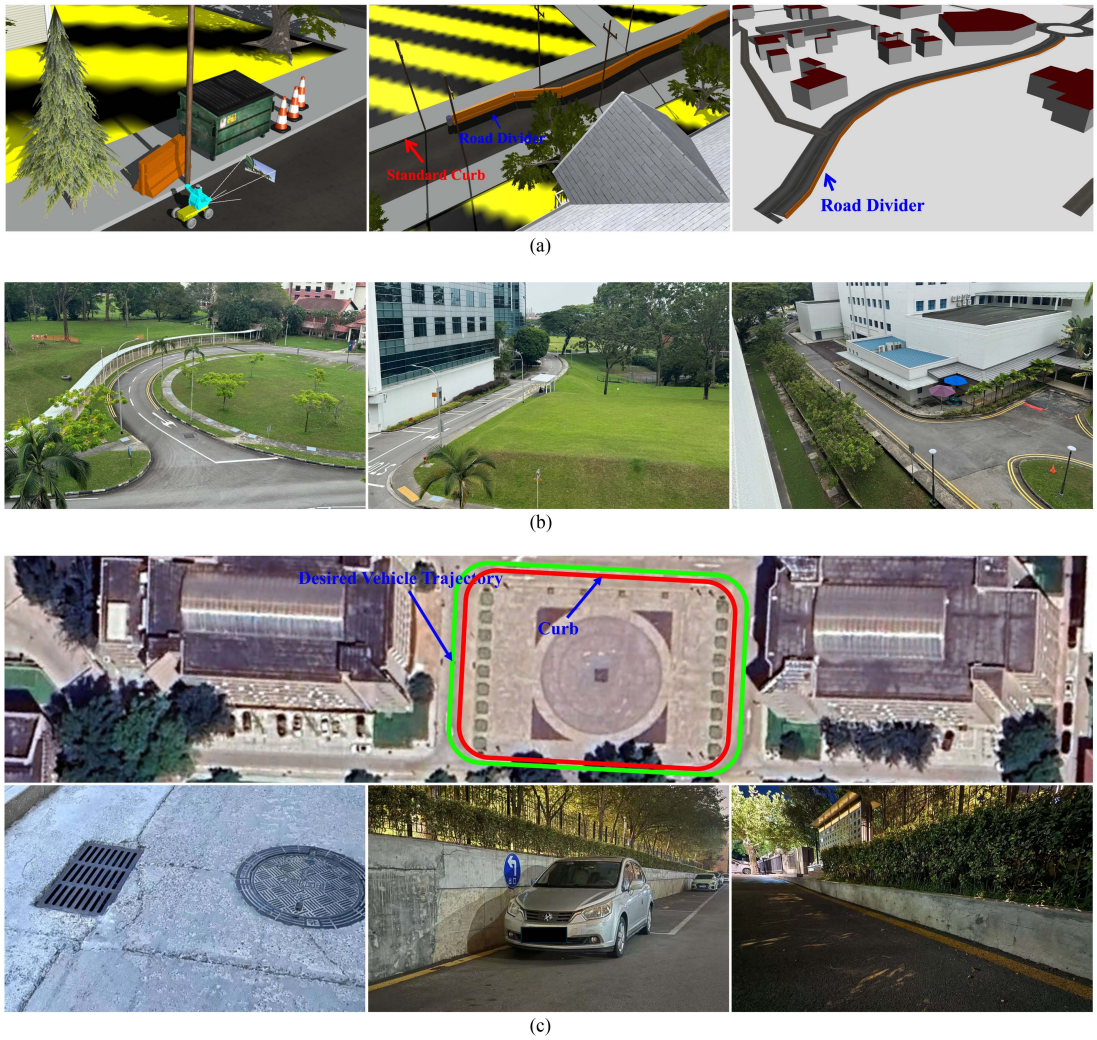


Fig. 5. Test scenarios in Gazebo and real world. (a) The test scenarios in Gazebo simulator. From left to right: Scenario 1, which includes various objects such as a tree, a utility pole with lamps, a bench, a trash bin, and traffic cones positioned near the curb. Scenario 2, which includes a standard curb and a road divider. Scenario 3, which models urban road environments with standard and road divider curb structure. (b) The test scenarios at NTU Campus. From left to right: Hall 7, which features a combination of concave and convex curb sections bordered by grassy terrain and trees. SPMS, which highlights a curved road surrounded by grassy terrain and trees, with a standard curb clearly delineating the roadway. WKW School, which showcases a straight roadway bordered by concrete curb. (c) Test scenarios on the SDU Campus for comprehensive evaluation of the proposed curb-following system. Top row: Aerial view of the experimental square, featuring a continuous circular curb used as the tracking reference. The autonomous vehicle was required to operate along the curb over extended periods. The road surface includes uneven segments, providing a challenging environment for both curb detection and curb-aligned motion generation. Bottom row (left): Close-up view of the square's road surface condition, illustrating irregular pavement features. Bottom row (middle): Scenario for evaluation of curb-following performance in high-curvature segments with geometric discontinuities, where the curb line is interrupted by a parked vehicle, requiring the system to execute a smooth detour and then resume normal curb following. Bottom row (right): Scenario for evaluation of curb-following performance on height-degrading curb segments, where the curb gradually transitions in height until becoming nearly flush with the road surface, testing the system's robustness under visually degraded curb conditions.

$$\begin{aligned} \dot{v} &= a \\ \dot{\delta} &= \Delta\delta \\ \dot{v}_s &= a_s \end{aligned} \tag{11}$$

where s denotes the travel progress along the curb curve of the virtual target point, v_s is the speed of the virtual target point, a denotes the vehicle's acceleration, $\Delta\delta$ is the rate of change of the steering angle, and a_s corresponds to the acceleration of the virtual target point. The augmented state vector is defined as $\boldsymbol{\xi} = [x\ y\ \theta\ s\ v\ \delta\ v_s]^\top$, while the augmented control input vector is expressed as $\bar{\boldsymbol{u}} = [a\ \Delta\delta\ a_s]^\top$.

2) Control Strategy: To apply the MPCC-based approach to the curb-aligned motion generation problem, we define the components of the multiobjective optimization problem as follows.

a) Tracking error minimization: The primary objective of the curb-following task is to maintain a specified lateral offset from the curb curve. Thus, the difference between the desired lateral offset l and the contour error e_k^c should be minimized. The cost term for tracking error minimization is formulated as

$$J_{c_k} = q_c \left\| \hat{e}_k^c(\boldsymbol{p}_{v_k}, s_k) - l \right\|^2 \tag{12}$$

where q_c is the weight parameter.

IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2026, Vienna, Austria. Cite as T-RO paper.

b) Lag error minimization and travel progress maximization: The lag error term is introduced to evaluate the accuracy of the approximation of the optimal solution for the contour error. In addition, another objective of the curb-following task is to ensure efficient task execution. To achieve this, the travel progress along the curb curve should be maximized. The cost term for the lag error minimization and travel progress maximization is defined as

$$J_{l_k} = q_l \left\| \hat{e}_k^l(\boldsymbol{p}_{v_k}, s_k) \right\|^2 - \gamma v_{s_k} \Delta T \tag{13}$$

where q_l is the weight of the lag error, and γ is the weight assigned to the travel progress. By penalizing these terms, the accuracy of the approximation and the efficiency of task execution can be guaranteed simultaneously.

c) Control input regularization: To ensure smooth vehicle operation, it is essential to minimize the rate of change of the control inputs. Failing to do so may result in significant fluctuations, potentially leading to instability in the vehicle's motion. The control input regularization cost is defined as

$$J_{\bar{u}_k} = \bar{\boldsymbol{u}}_k^\top R_{\bar{u}} \bar{\boldsymbol{u}}_k \tag{14}$$

where $R_{\bar{u}} \in \mathbb{R}^{3\times 3}$ is the weight matrix of the rate of change of the control inputs.

Based on the formulations given above, the MPCC formulation of the curb-aligned motion generation problem is defined as

$$\begin{aligned}
&\min_{\bar{\boldsymbol{u}}} \sum_{k=0}^{N} J_{c_k} + J_{l_k} + J_{\bar{u}_k} \\
&\text{s.t.} \quad \boldsymbol{\xi}_0 = \boldsymbol{\xi} \\
&\quad \boldsymbol{\xi}_{k+1} = f(\boldsymbol{\xi}_k, \bar{\boldsymbol{u}}_k), \quad k = 0, \ldots, N-1 \\
&\quad \boldsymbol{\xi}^- \le \boldsymbol{\xi}_k \le \boldsymbol{\xi}^+, \quad k = 1, \ldots, N, \\
&\quad \bar{\boldsymbol{u}}^- \le \bar{\boldsymbol{u}}_k \le \bar{\boldsymbol{u}}^+, \quad k = 0, \ldots, N-1
\end{aligned} \tag{15}$$

which is a nonlinear optimal control problem. To enable real-time implementation, the nonlinear terms in the problem are approximated at each time step k using a second-order Taylor expansion

$$J(\boldsymbol{\xi}) \approx J(\boldsymbol{\xi}_0) + J'(\boldsymbol{\xi}_0)(\boldsymbol{\xi} - \boldsymbol{\xi}_0) + \frac{J''(\boldsymbol{\xi}_0)}{2!}(\boldsymbol{\xi} - \boldsymbol{\xi}_0)^2 \tag{16}$$

where the current observation $\boldsymbol{\xi} = \boldsymbol{\xi}_0$ is used as the first linearization point, and subsequent linear points for later control horizons are calculated using results from the previous optimization. This approximation transforms the nonlinear optimization problem into a local quadratic programming (QP) problem as follows:

$$\begin{aligned}
&\min_{\boldsymbol{\xi}, \bar{\boldsymbol{u}}} \quad \sum_{k=0}^{N} \boldsymbol{\xi_k}^\top \mathbf{Q} \boldsymbol{\xi_k} + \mathbf{q}^\top \boldsymbol{\xi_k} + \bar{\boldsymbol{u}}_k^\top \mathbf{R}_{\bar{\mathbf{u}}} \bar{\boldsymbol{u}}_k + \mathbf{c}^\top \bar{\boldsymbol{u}}_k - \gamma v_{s_k} \Delta T \\
&\text{s.t.} \quad \boldsymbol{\xi_0} = \boldsymbol{\xi} \\
&\quad \boldsymbol{\xi}_{k+1} = A_k \boldsymbol{\xi}_k + B_k \bar{\boldsymbol{u}}_k + g_k \\
&\quad \boldsymbol{\xi}^- \le \boldsymbol{\xi}_k \le \boldsymbol{\xi}^+, \quad k = 1, \ldots, N
\end{aligned}$$

Algorithm 1: The proposed MPCC-based curb-aligned motion generation method.

Input: ${}^c p(s)$ the cubic spline curb curve, $\boldsymbol{x}_{\text{init}}$ initial state obtained from odometry

$s_0 \leftarrow$ compute initial travel progress from cubic spline curve ${}^c p(s)$ using $\boldsymbol{x}_{\text{init}}$
loop
  if not initialized **then**
    $\boldsymbol{\xi}_0 \leftarrow 0$, $\bar{\boldsymbol{u}}_{0:N-1} \leftarrow 0$
    $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_{\text{init}}$
    for $k \in [1, N]$ **do**
      $s_k = s_{k-1} + v_{\text{initial}} \Delta T$
      $\boldsymbol{\xi}_k \leftarrow$ compute reference augmented state vector from cubic spline curb curve ${}^c p(s)$ using s_k
    end for
  else
    for $k \in [1, N-1]$ **do**
      $\boldsymbol{\xi}_{k-1} \leftarrow$ from previous $\boldsymbol{\xi}^*_{\text{pred},k}$
      $\bar{\boldsymbol{u}}_{k-1} \leftarrow$ from previous $\bar{\boldsymbol{u}}^*_{\text{pred},k}$
    end for
    $\boldsymbol{\xi}_N \leftarrow$ computing from forward simulation using $\boldsymbol{\xi}_{N-1}$
  end if
  Compute (12) and (13) given $\boldsymbol{\xi}_k$
  $\bar{\boldsymbol{u}}^*_{\text{pred},0:N-1}, \boldsymbol{\xi}^*_{\text{pred},1:N} \leftarrow$ solve (15)
  Apply v_0^* and δ_0^* to the hardware platform
end loop

$$\bar{\boldsymbol{u}}^- \le \bar{\boldsymbol{u}}_k \le \bar{\boldsymbol{u}}^+, \quad k = 0, \ldots, N-1 \tag{17}$$

where $\boldsymbol{\xi}_k$ and $\bar{\boldsymbol{u}}_k$ represent the augmented state and control input vectors at time step k, respectively; $\mathbf{Q}$, $\mathbf{R}_{\bar{\mathbf{u}}}$, $\mathbf{q}$, and $\mathbf{c}$ are weight matrices; A_k, B_k, and g_k are system matrices; and $\underline{\boldsymbol{\xi}}$, $\bar{\boldsymbol{\xi}}$, $\underline{\bar{\boldsymbol{u}}}$, and $\bar{\bar{\boldsymbol{u}}}$ denote the lower and upper bounds on state and control inputs, respectively. The optimal control inputs are obtained by iteratively solving the QP problem using the sequential quadratic programming (SQP) approach. We use the operator splitting quadratic program (OSQP) solver [52] to solve the QP problem. The proposed MPCC-based curb-aligned motion generation method is summarized in Algorithm 1.

IV. Experimental Results

To validate the proposed curb-following system, we conducted experiments in both simulations and real-world environments. Simulations were carried out in the Gazebo simulator, while real-world experiments were conducted on public roads within the campus of Nanyang Technological University (NTU) and the campus of Shandong University (SDU). The proposed system was implemented on an Hunter 2.0, a four-wheel Ackerman-steering mobile robot. In Gazebo simulations, the robot was equipped with a Robosense RS-LiDAR-32 (as shown in the left part of Fig. 6), while in real-world experiments, the robot was equipped with an Ouster OS1-128 LiDAR on the NTU Campus (as shown in the middle part of Fig. 6) and an Ouster OS1-64 LiDAR on the SDU Campus (as shown in the right part

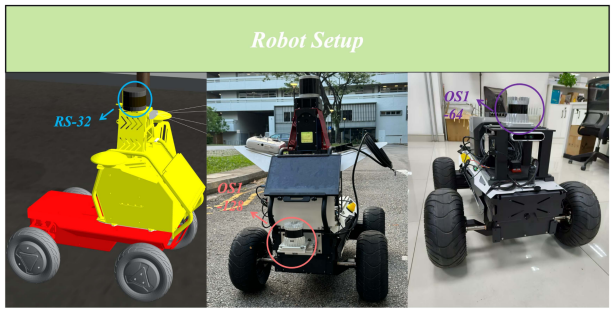


Fig. 6. Experimental setup [Left: Hunter 2.0 equipped with RS-32 in Gazebo simulations. Middle: Hunter 2.0 equipped with OS1-128 LiDAR in real-world experiment (NTU Campus). Right: Hunter 2.0 equipped with OS1-64 LiDAR in real-world experiment (SDU Campus).]

TABLE I
KEY PARAMETERS

Symbol	Definition	Value
N_s	The number of sampled points at each frame	50
N	The prediction horizon	40
r	The resolution of elevation map	0.05m
l	The desired normal deviation from curb	0.8m
q_c	The weight of contour error	10.0
q_l	The weight of lag error	2.0
γ	The weight of travel progress	0.5

of Fig. 6). The proposed system was programmed in C++ within the robot operating system, Noetic release, running on a laptop equipped with an AMD Ryzen 7 5800H CPU (3.2 GHz) and 16 GB RAM. The curb detection module ran at 15 Hz, and the curb-aligned motion generation module ran at 50 Hz. The key parameters used in the experiments are listed in Table I.

A. Evaluation Metrics

To quantitatively evaluate the performance of the proposed curb-following system, ground truth data from the MCD dataset [53] was utilized. The MCD dataset provides a high-accuracy, survey-grade prior point cloud map of NTU Campus, along with the ground truth curb points (as shown in Fig. 7). During real-world experiments, we employed HDL-localization [54], a map-based localization module, to estimate the vehicle's pose within the high-accuracy point cloud map.

The performance of the proposed system was evaluated using three key metrics: 1) real-time tracking error; 2) mean tracking error; and 3) task completion time. The real-time tracking error measures the lateral deviation between the vehicle's current position and the ground truth curb at each moment, while the mean tracking error quantifies the average deviation over the entire task duration. This evaluation method is effective because the performance of the curb-aligned motion generation module inherently reflects the combined accuracy of curb detection and curb-aligned motion generation. When both the real-time and mean tracking errors are close to the desired lateral offset, it indicates that the curb has been accurately detected and the vehicle is successfully following it. The task completion time is used to assess the efficiency of the system in completing the curb-following task. Therefore, achieving a low tracking error alongside a short time cost provides a comprehensive measure of the overall effectiveness and efficiency of the proposed curb-following system.

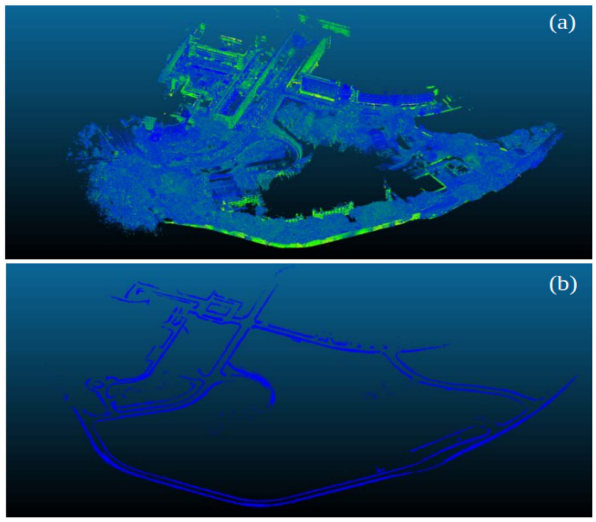


Fig. 7. Ground truth of NTU Campus [(a) High-accuracy survey-grade prior map of NTU Campus. (b) Ground truth curb.].

Let the real-time tracking error be denoted as E_k and the mean tracking error as $\bar{E}$

$$E_k = \|e_k^c - l\|$$
$$\bar{E} = \frac{1}{N_{\text{step}}} \sum_{k=1}^{N_{\text{step}}} \|e_k^c - l\| \tag{18}$$

where N_{step} represents the total number of time steps in the task. The task completion time t is defined as the total execution time.

To compute the lateral offset from the ground truth curb, a local subset of ground truth curb points is extracted around the vehicle at each time step. This subset is then fitted to a cubic spline curve, which serves as the reference curb. The projection point of the current position of the vehicle onto this reference curve is determined to calculate the lateral offset for each time step, thereby deriving the real-time tracking error. In addition to tracking accuracy, the efficiency of task execution is assessed by analyzing the vehicle's speed profile and execution time. The vehicle's speed is recorded at each time step, while the total execution time is measured at the conclusion of the task.

B. Safety Considerations

To ensure the robustness and safety of the proposed framework in real-world applications, we incorporate multiple mechanisms to address potential failures in both curb detection and trajectory tracking.

1) *Curb Detection Safety:* The curb detection module includes a confidence evaluation mechanism that assesses the temporal consistency of curb estimates across consecutive frames. Specifically, we compute the differences in

curvature $\Delta\kappa$ and lateral offset Δd between detected curb segments in two consecutive frames. If either condition is met

$$|\kappa_t - \kappa_{t-1}| > \kappa_{\text{thresh}} \quad \text{or} \quad |d_t - d_{t-1}| > d_{\text{thresh}} \tag{19}$$

the curb detection is considered unreliable. In such cases, the system either switches to the opposite curb (if available) or issues a stop command to prevent unsafe behavior. This mechanism ensures that inconsistent or abrupt detections do not propagate to downstream modules, thereby avoiding unstable motion.

2) *Curb-Aligned Motion Generation Safety:* Safety in trajectory tracking is maintained through a combination of hard constraints and runtime evaluation. The MPCC-based controller enforces strict bounds on steering angle, acceleration, and other dynamic constraints to ensure trajectory feasibility. Extensive simulations show that the maximum detection error is bounded by the elevation map resolution (e.g., 0.05 m), while the maximum tracking error is approximately 0.09 m. Based on this, we define a dynamic safety threshold, and if the tracking error exceeds the following condition:

$$\hat{e}_k^{\text{track}} > |\hat{e}_{\max}^{\text{track}} - \hat{e}_{\max}^{\text{detect}}| \tag{20}$$

the system triggers an emergency stop to prevent the vehicle from deviating beyond the safe drivable area. In addition, if the QP solver fails to return a feasible solution within the allowed time, the system also initiates a safe stop. These mechanisms ensure operational safety under uncertain and potentially degraded environmental conditions.

These layered safety strategies provide strong safeguards for autonomous curb-following tasks, enhancing system reliability in complex and dynamic real-world environments.

C. Ablation Studies in Simulation

To demonstrate the robustness and effectiveness of the proposed methods, we conducted ablation studies in simulation to evaluate the complete curb-following system by comparing the modules with their corresponding variants. Based on the key components of the proposed system, we designed the following ablation settings.

1) *Curb Detection*: The proposed curb detection method, comprising *adaptive curb feature extraction* and *robust curb point association*, is replaced by the curb point extraction methods described in [27] (Method A) and [55] (Method B). A qualitative analysis is conducted to compare the performance of the different methods, focusing on their robustness and adaptability across various road conditions.

2) *Curb-Aligned Motion Generation*: The proposed MPCC-based curb-aligned motion generation method is replaced by the constant speed linear quadratic regulator (CLQR) controller and the constant speed nonlinear MPC (CNMPC) described in [56]. The performance of the three methods is evaluated using the real-time tracking error E_k and the mean tracking error $\bar{E}$. In addition, execution time and the speed profile of the vehicle are analyzed to assess the efficiency and dynamic behavior of the methods.

1) Evaluation Scenarios: We designed three test scenarios for the ablation studies in simulation, as shown in Fig. 5(a).

- *Scenario 1*: This scenario simulates a road environment with various roadside features, including a tree, a utility pole with lamps, a bench, a trash bin, and traffic cones [see Fig. 5(a), left]. These elements are positioned near the curb, creating curb-like features that could potentially interfere with curb detection. The goal of this scenario is to evaluate the robustness of the curb detection algorithm in distinguishing actual curbs from surrounding distractions.
- *Scenario 2*: This scenario tests the system's capability to handle curbs of varying heights and types, including standard curbs and road divider curbs. It is also used for ablation studies of the curb-aligned motion generation controller. As shown in Fig. 5(a), middle, this scenario features transitions between different curb types and includes turns with varying curvatures. The scenario provides a comprehensive test of the system's capability to adapt to varying curb types and the robustness in navigating complex road geometries.
- *Scenario 3*: This scenario features a 200-m road section with a road divider, as depicted in Fig. 5(a), right. The primary objective is to evaluate the system's performance over long distances. The robot is tasked with exclusively tracking the road divider while navigating sections with varying curvatures. This scenario presents a challenging environment to assess the system's ability to maintain stable and accurate curb following over long distances, ensuring consistent performance in real-world applications.

2) Ablation Study for Curb Detection: This section presents the simulation results of curb detection in *Scenario 1* and *Scenario 2*. In each experimental scenario, the vehicle is required to drive from the same predefined starting point to a designated goal location. The detection results at each frame are visualized in Fig. 9. Given that the resolution of the elevation map in our experiment setup is 0.05 m, we define a successful detection as a frame in which more than 75% of the detected curb points lie within 0.05 m of the ground truth curb location; otherwise, the detection is considered a failure. The first two rows of Fig. 8 illustrate the curb points extracted by the proposed method compared to the baseline methods in *Scenario 1*. In scenarios without roadside feature interference (Fig. 8, top row and Fig. 9, top row), all three methods (Method A, Method B, and the proposed method) successfully detect curb points. The baseline methods effectively identify curb features and extract curb points, achieving performance comparable to the proposed method in this simple environment.

However, the difference in performance becomes apparent in the more challenging scenario with roadside feature interference (Fig. 8, middle row and Fig. 9, top row). This scenario includes objects, such as a tree, a utility pole with lamps, a bench, a trash bin, and traffic cones positioned near the curb, introducing significant noise and occlusions. Under these conditions, the baseline methods, which rely on predefined thresholds, fail to adapt to the dynamic environment. The interference caused

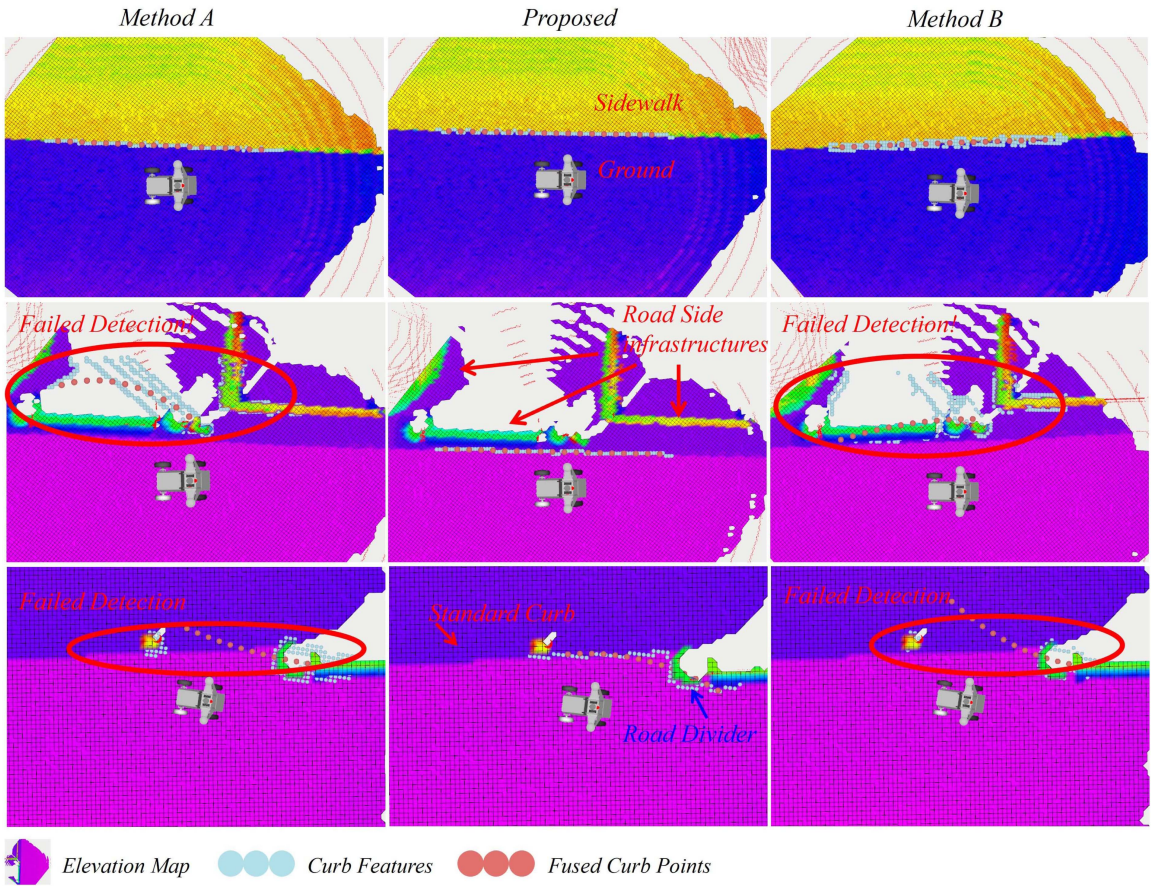


Fig. 8. Results of the ablation study for curb detection in *Scenario 1* and *Scenario 2*. [Top row: All three methods successfully detected curb points in the simple road environment without the interference from roadside infrastructures. Mid row: The baseline methods failed to detect curb points due to the interference from roadside infrastructures, while the proposed method successfully detected them, demonstrating its adaptability to challenging environments. Bottom row: The result of the ablation study in *Scenario 2*. The proposed method successfully detected both standard curbs and road dividers, while baseline methods failed to detect curb points due to their inability of handling diverse curb dimensions and types, as well as environmental challenges.]

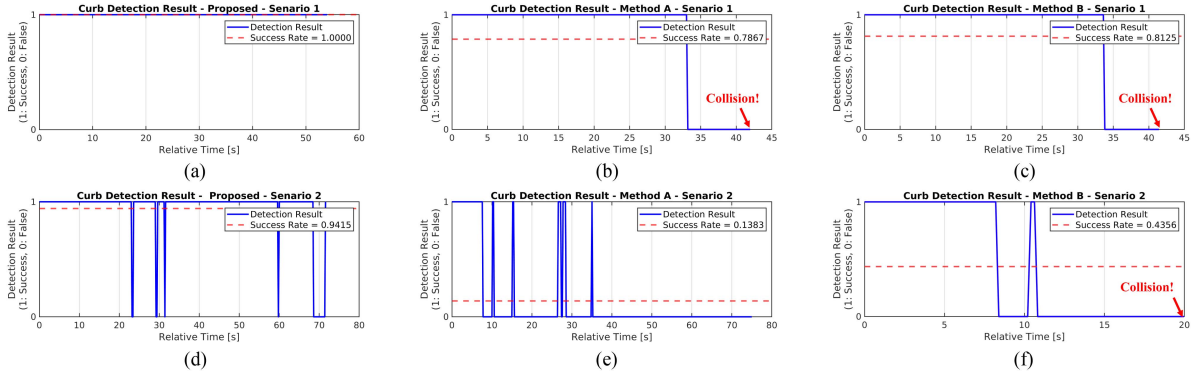


Fig. 9. Comparison of detection results over extensive sequences of frames between the proposed method and baseline methods in *Scenario 1* and *Scenario 2*. (a) Detection result (Proposed, Scenario 1). (b) Detection result (Method A, Scenario 1). (c) Detection result (Method B, Scenario 1). (d) Detection result (Proposed, Scenario 2). (e) Detection result (Method A, Scenario 2). (f) Detection result (Method B, Scenario 2).

by nearby objects leads to misclassifications, preventing these methods from distinguishing the curb from surrounding noise. Due to these detection failures, the vehicle collides with the curb during navigation. In contrast, the proposed method demonstrates robustness and adaptability by accurately extracting curb points despite the presence of complex roadside features. This superior performance highlights the effectiveness of the proposed method in handling dynamic and challenging real-world conditions, showcasing its reliability in scenarios with significant environmental interference.

The results shown in Fig. 8, bottom row and Fig. 9, bottom row compare the performance of curb detection methods in

TABLE II
ABLATION STUDY FOR CURB DETECTION

Scenario	Method	Success Rate(%)	Task Finished
Scenario 1	Proposed	**100**	**Yes**
	Method A	78.67	No
	Method B	81.25	No
Scenario 2	Proposed	**94.15**	**Yes**
	Method A	13.83	No
	Method B	43.56	No

Scenario 2. The baseline methods fail to detect curb points, particularly during transitions between the curb and the road divider. Their reliance on predefined thresholds significantly limits their adaptability to varying road conditions, resulting in detection failures in these complex transitions. Although Method A did not lead to a collision, the vehicle deviated from the expected lateral shift distance due to inaccurate detection. In the case of Method B, the detection failure resulted in a collision with the curb. In contrast, the proposed method consistently detects curb points throughout this scenario, demonstrating robust performance across diverse curb types and elevation variations. This highlights the method's adaptability and reliability in handling complex road environments with dynamic and changing curb features. The results demonstrate the superior generalization capability of the proposed method, making it well-suited for real-world applications with varying curb characteristics.

Table II summarizes the ablation study results evaluating the effectiveness of different curb detection methods across two representative scenarios. The table reports the Success Rate (%), defined as the percentage of frames in which the curb detection was successful, and whether the curb-following task was successfully completed (Task Finished). As shown, the proposed method achieves the highest success rates in both scenarios—100% in Scenario 1 and 94.15 % in Scenario 2—demonstrating superior detection accuracy and robustness. In contrast, both baseline methods perform significantly worse, particularly in Scenario 2, where their low success rates and failure to complete the task highlight their limited adaptability in complex environments. This comparison clearly validates the advantages of the proposed method in terms of both perception accuracy and practical navigation capability.

3) Ablation Study for Curb-Aligned Motion Generation: To benchmark the performance of the proposed MPCC-based curb-aligned motion generation method, *Scenario 2* and *Scenario 3* were utilized, comparing it against the CLQR method and the CNMPC method [56]. In the simulation setup, both the CLQR and CNMPC methods were designed to follow the curb curve at a constant speed of 1.0 m/s, while the maximum speed of the MPCC-based curb-aligned motion generation method was also set to 1.0 m/s, as shown in Fig. 12. The vehicle was required to maintain a normal deviation of 0.8 m while following the curb. Screenshots of these two scenarios are shown in Fig. 10.

As shown in Fig. 11(a)–(c), the MPCC-based curb-aligned motion generation method consistently achieves lower real-time tracking errors and the lowest mean tracking error

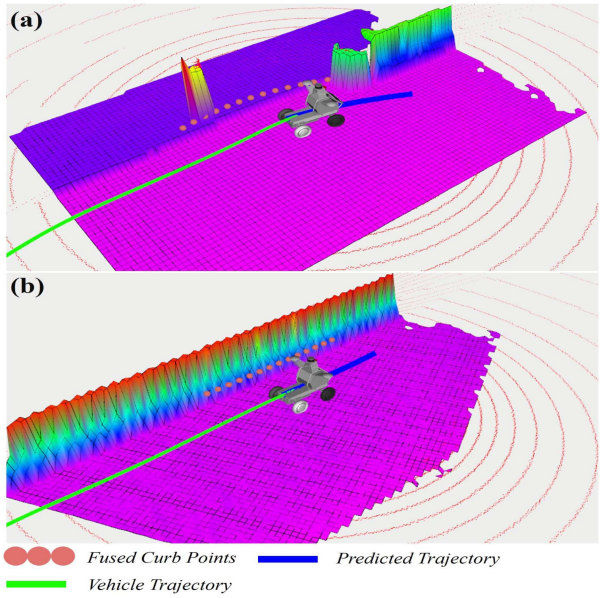


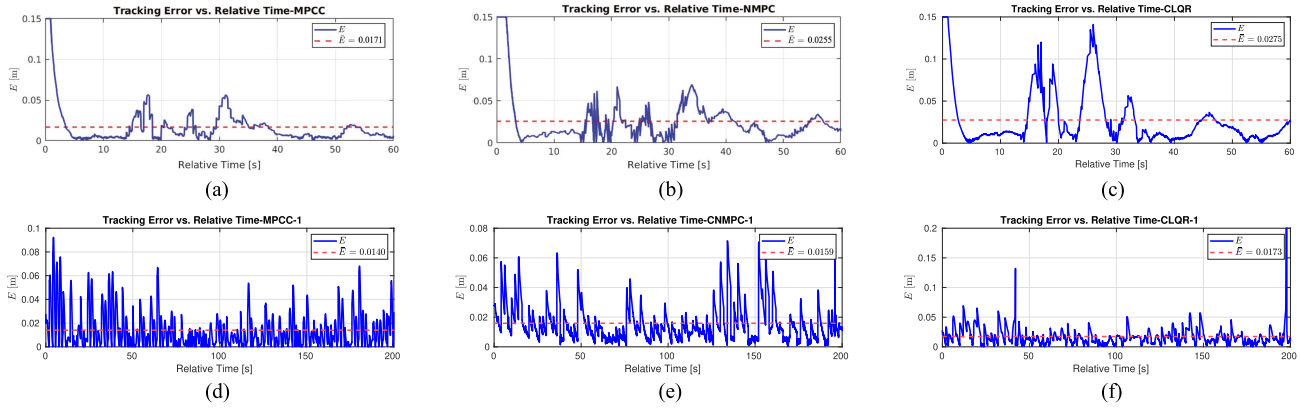
Fig. 10. Screenshots of the proposed MPCC-based curb-aligned motion generation method in simulation *Scenario 2* [Fig(a)] and *Scenario 3* [Fig(b)]. [The red points represent fused curb points, the blue line indicates the predicted trajectory generated by the proposed method, and the green line depicts the actual vehicle trajectory.].

TABLE III
ABLATION STUDY FOR CURB-ALIGNED MOTION GENERATION

Scenario	Method	$\bar{E}(m)$	$t(s)$
Scenario 2	Proposed	**0.0171**	62.64
	CNMPC	0.0255	61.95
	CLQR	0.0275	**61.87**
Scenario 3	Proposed	**0.0140**	202.88
	CNMPC	0.0159	202.48
	CLQR	0.0173	**202.46**

($\bar{E} = 0.0171$ m) in *Scenario 2*, showcasing its superior curb-following performance. The larger tracking errors observed with the CLQR and CNMPC methods can be attributed to their inability to adjust speed dynamically during turns. Both of them maintain a constant high speed, resulting in significant deviations from the desired path in curved sections. In contrast, the MPCC-based curb-aligned motion generation method dynamically optimizes the vehicle's speed based on the curb curvature, reducing speed when necessary to ensure accurate curb following and minimize tracking errors. This adaptive speed control enables the MPCC-based curb-aligned motion generation method to excel in complex road environments, providing precise and reliable curb-aligned motion. Furthermore, as shown in Table III, the MPCC-based curb-aligned motion generation method completes the task in 62.64 s, which is comparable to CNMPC (61.95 s) and CLQR (61.87 s), with only a 0.77-s difference from the fastest method.

The tracking performance of the three methods in *Scenario 3* is illustrated in Fig. 11(d)–(f). Here, the MPCC-based curb-aligned motion generation method again outperforms the others, achieving a mean tracking error of 0.0140 m, the lowest among

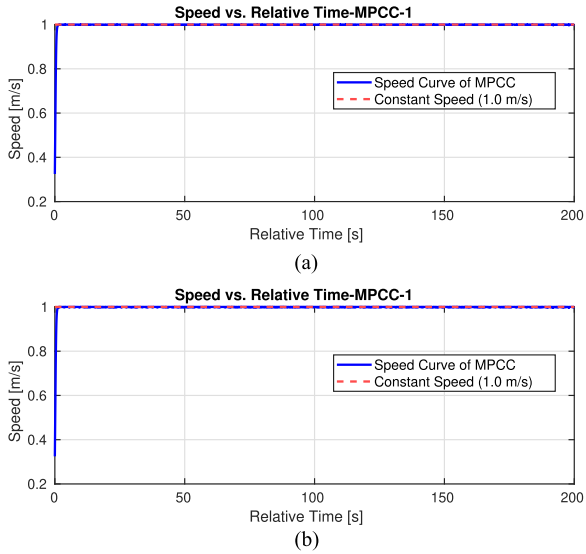


(a) (b) (c) (d) (e) (f)

Fig. 11. Real-time tracking error and the mean tracking error (read dashed line) of three curb-aligned motion generation methods in simulation. (a) Tracking error (Proposed, Scenario 2). (b) Tracking error (CNMPC, Scenario 2). (c) Tracking error (CLQR, Scenario 2). (d) Tracking error (Proposed, Scenario 3). (e) Tracking error (CNMPC, Scenario 3). (f) Tracking error (CLQR, Scenario 3).

the three methods. Its real-time tracking error remains consistently small throughout the simulation, demonstrating stable and precise trajectory tracking. In comparison, the CNMPC method has a slightly higher mean tracking error of 0.0159 m, with moderate fluctuations and occasional spikes, indicating reduced stability. The CLQR method, with the highest mean tracking error of 0.0173 m, exhibits larger fluctuations and more frequent spikes, reflecting lower precision and stability in tracking. These results highlight the superior performance of the proposed method in maintaining accurate and stable curb-aligned motion generation in *Scenario 3*. In addition, as shown in Table III, the execution time of the proposed curb-aligned motion generation method method differs by only 0.42 s compared to the fastest method, CLQR, further demonstrating its high task completion efficiency.

In summary, the MPCC-based curb-aligned motion generation method achieves exceptional tracking accuracy while maintaining high task completion efficiency, making it a robust and reliable solution for curb-aligned motion generation in complex road environments. In contrast, the baseline methods struggle under the challenging road conditions, resulting in a high risk of collision to the curb. If the baseline methods are used in the real-world experiments, the vehicle must slow down to avoid collision, resulting in a longer task completion time.



(a) (b)

Fig. 12. Speed profiles generated by the proposed MPCC-based curb-aligned motion generation method (blue line) and baseline methods (red dashed line) in Scenario 2 and Scenario 3. (a) Speed profile of the vehicle in Scenario2 0. (b) Speed profile of the vehicle in Scenario3.

D. Real-World Experiments

To further validate the performance and effectiveness of the proposed system, real-world experiments were carried out within NTU Campus (Fig. 5, middle) and SDU Campus (Fig. 5, bottom). The Hunter 2.0 robot autonomously drove along curbs based on online curb detection. To reduce computational load, the point cloud data generated by the Ouster 128-line LiDAR sensor was downsampled to 32 lines. In these experiments, the system was assigned to detect curbs and maintain a normal deviation of 0.8 m from the curb.

1) Real-World Experiments on the NTU Campus: On the NTU Campus experiments, four scenarios were considered, as shown in Fig. 5(b).

The first experiment was conducted on public roads near the School of Physical and Mathematical Sciences (SPMS), featuring mildly curved urban roads with a mixture of roadside infrastructures, vegetation, and curbs. These conditions provided a classical urban environment to test the system's performance in curb detection and curb-aligned motion generation. The trajectory of the vehicle and the detected curb are shown in Fig. 13(a), top. As shown in the tracking error profile [Fig. 14(a)], the system achieved a mean tracking error of 0.0299 m, demonstrating precise alignment with the curb in these urban road conditions.

The second experiment took place near the Wee Kim Wee School of Communication and Information (WKW). The roads in this area were relatively straight with fewer obstructions, offering a simpler urban environment to evaluate the stability and

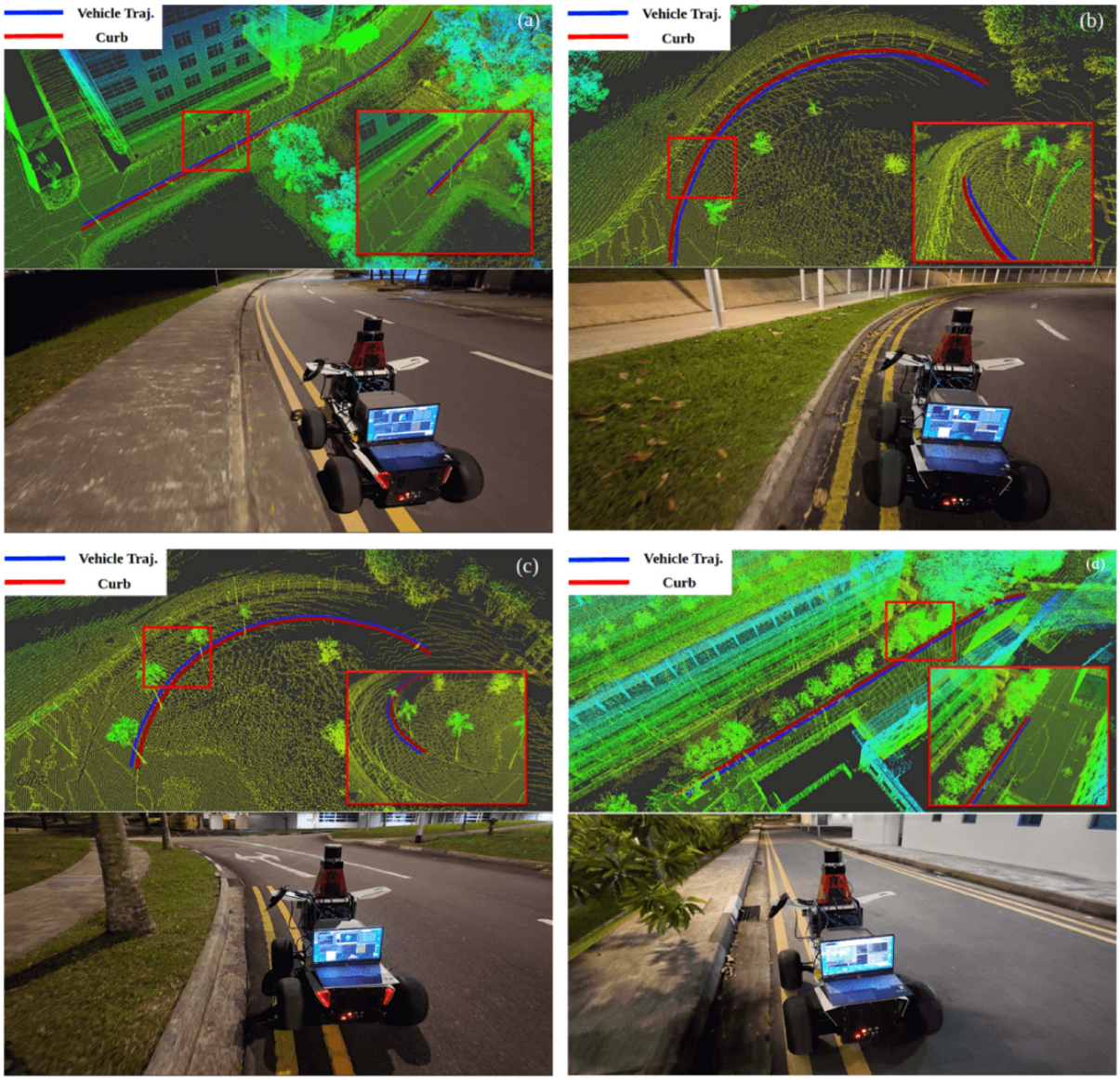


Fig. 13. Real-world experiments conducted on public roads at NTU Campus. For subfigures (a)– (d), the top row displays the visualization in RViz, while the bottom row presents corresponding on-site photographs. (a) SPMS. (b) Hall 7 - convex side curb. (c) Hall 7 - concave side curb. (d) WKW School of Communication and Information.

accuracy of the system. The trajectory and curb detection results are shown in Fig. 13(d). Despite the relatively straightforward road geometry, the tracking error profile [Fig. 14(b)] indicates a mean tracking error of 0.0391 m, reflecting consistent and reliable curb-following performance even in less challenging scenarios.

The third and fourth experiments were conducted on roads near Hall 7, specifically in concave and convex road sections [as shown in Fig. 13(b) and 13(c). These scenarios were selected to evaluate the system's performance in more complex urban environments. The roads in these areas are characterized by challenging features, such as curved curbs, dense landscaping, trees, and grass, which introduce noise and irregularities into the sensor data. These conditions test the system's ability to reliably detect and follow curbs in the presence of significant environmental variability.

In the concave road section near Hall 7 [Fig. 13(b)], the system maintained a mean tracking error of 0.0248 m, as shown in the tracking error profile [Fig. 14(c)]. Despite the presence of roadside interference, such as trees and grass, the system demonstrated stable performance, effectively aligning with the curb throughout the experiment.

Similarly, in the convex road section near Hall 7 [Fig. 13(c)], the system achieved a mean tracking error of 0.0291 m, as indicated in Fig. 14(d). While the environmental conditions introduced slight variations in tracking error compared to the concave scenario, the results highlight the system's robustness in handling environmental challenges.

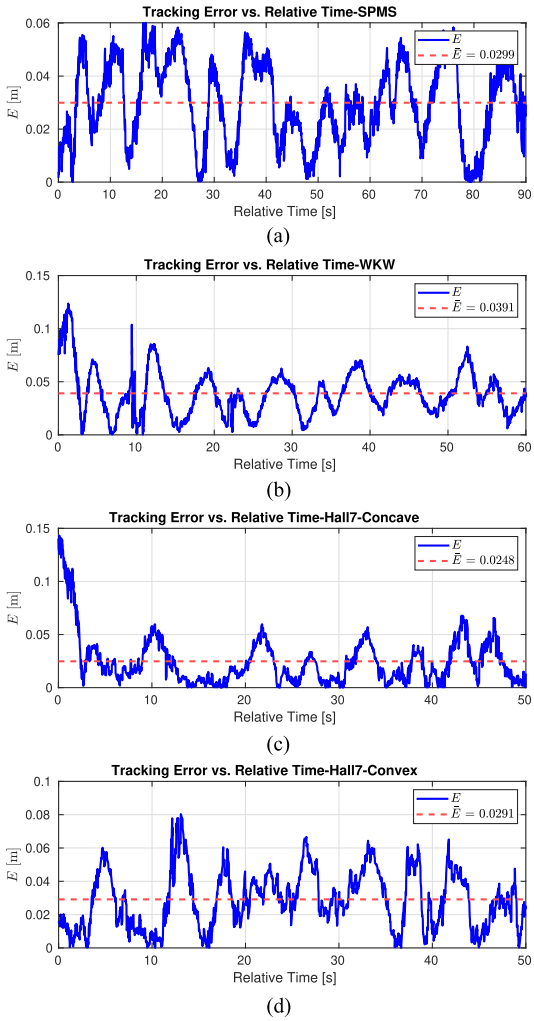


Fig. 14. Real-time tracking error of four different real-world scenarios on the NTU Campus. (a) Tracking error of real-world experiment in SPMS. (b) Tracking error of real-world experiment in WKW. (c) Tracking error of real-world experiment in Hall7-concave. (d) Tracking error of real-world experiment in Hall7-convex.

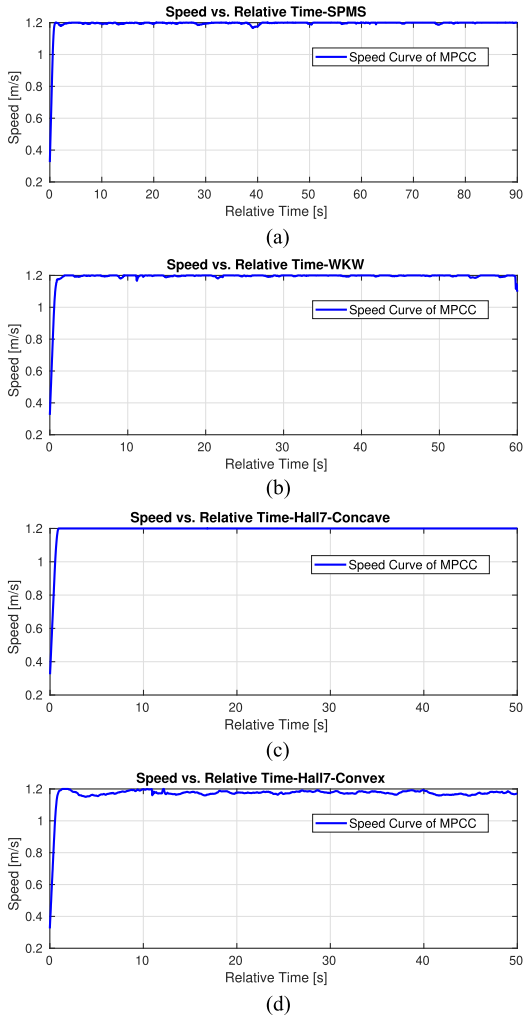


Fig. 15. Vehicle speed profiles of four different real-world scenarios on the NTU Campus. (a) Vehicle speed profile of real-world experiment in SPMS. (b) Vehicle speed profile of real-world experiment in WKW. (c) Vehicle speed profile of real-world experiment in Hall7-concave. (d) Vehicle speed profile of real-world experiment in Hall7-convex.

Furthermore, the speed profiles of the real-world experiments conducted in SPMS, WWK, and Hall 7 (both concave and convex sections) are shown in Fig. 15. Across all scenarios, the vehicle consistently maintained a high speed, with the maximum speed set to 1.2 m/s. Stable speed profiles indicate that the proposed system ensures efficient task execution.

2) Real-World Experiments on the SDU Campus: To validate the robustness and adaptability of the proposed curb-following system, we conducted real-world experiments across three representative and challenging scenarios on the SDU Campus. These tests evaluate the system's performance under uneven terrain, geometric discontinuities, and height-degrading curbs.

a) Long-term curb following in complex real-world terrain: This experiment was conducted in an open square area of SDU Campus [see Fig. 5(c), top row], where a continuous curb forms a loop around the square. The terrain includes uneven surfaces, such as height-varying tiles [Fig. 5(c), bottom row(right)], posing a challenging testbed for the proposed curb-following system. Fig. 16(a) shows the online map visualization in RViz during the experiment, alongside photographic documentation of the test scenario.

The primary goal of this experiment was to evaluate the long-term operational stability of the proposed system and assess its robustness and adaptability to diverse, previously unseen terrains. During the experiment, the vehicle autonomously followed the curb loop for multiple laps. Despite the presence of surface irregularities, the system consistently maintained accurate curb following. The tracking error consistently remained within acceptable bounds throughout the experiment, with a mean value of 0.0185 m [as shown in Fig. 17(a)], demonstrating the robustness and adaptability of the proposed method to real-world surface variations. The speed of the vehicle remained stable throughout the experiment, with a maximum speed of 1.2 m/s [Fig. 18(a)], reflecting efficient and consistent control. These results demonstrate that the proposed system's capability for reliable long-duration operation in complex and unstructured road environments, underscoring its practical applicability for real-world deployment.

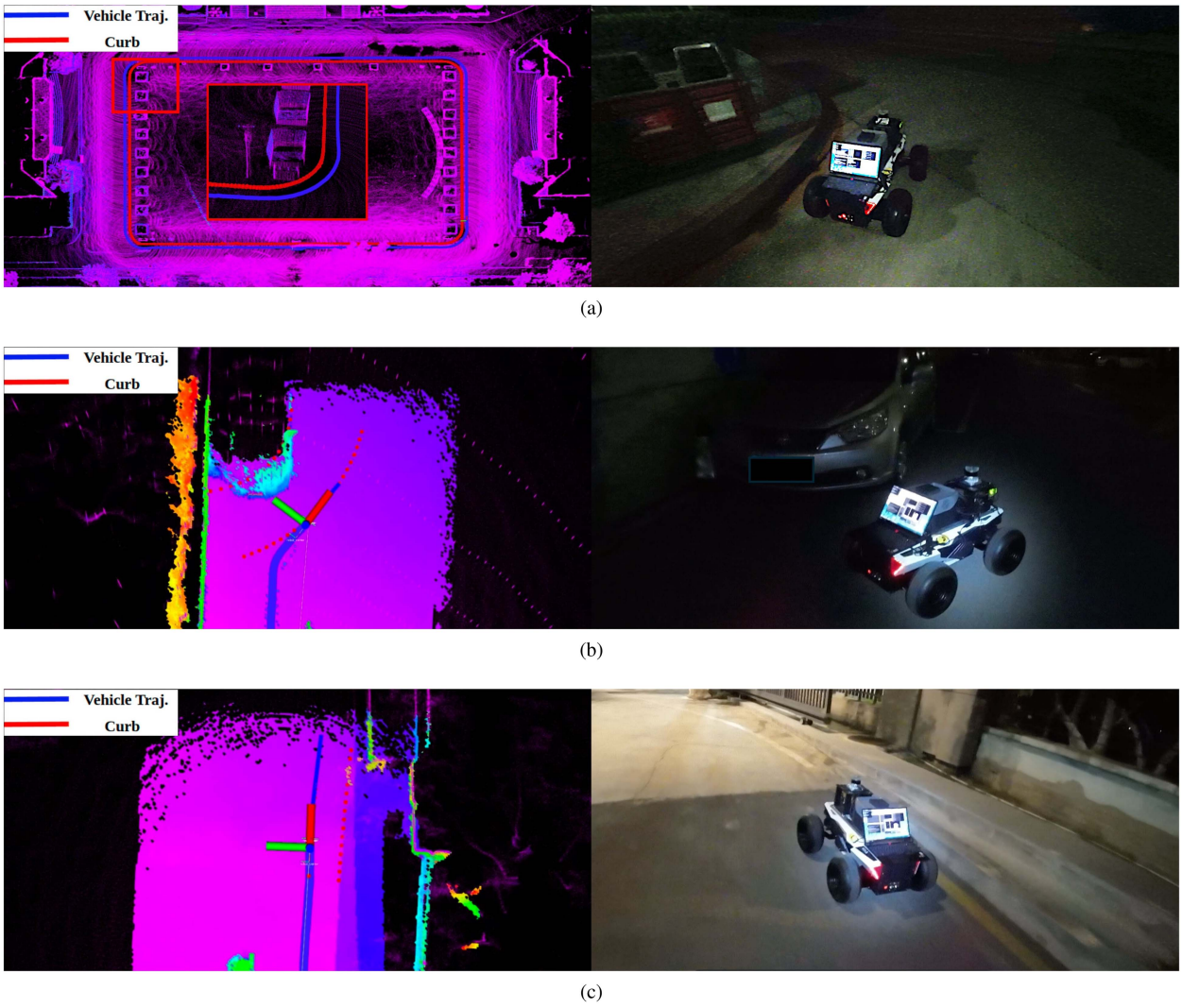


Fig. 16. Real-world experiments conducted on the SDU Campus. (a) Experimental site in an open square area with unobstructed curb boundaries. (b) Curb segment partially occluded by a parked vehicle, inducing high curvature and discontinuities in the curb line. (c) Curb section with gradually varying height, resulting in visually degraded boundaries that challenge perception accuracy.

b) Curb following in high-curvature segments with geometric discontinuities: Experiments were conducted on a curved road segment on the SDU Campus [Fig. 5(c), bottom row (middle)], where a parked vehicle obstructed part of the curb, creating a geometric discontinuity. In this case, the system temporarily used the vehicle's contour as a substitute boundary before realigning with the curb after bypassing the obstacle. Fig. 16(b) presents the real-time curb detection and motion generation visualizations alongside photographic documentation.

The primary objective of this experiment was to assess the system's robustness in high-curvature environments with abrupt geometric changes and occlusions. During the experiment, the vehicle consistently performed smooth avoidance maneuvers and accurately reengaged with the curb, maintaining a mean tracking error of 0.0292 m [Fig. 17(b)]. The speed profile in Fig. 18(b) shows stable motion with a maximum of 0.5 m/s. These results confirm the system's effectiveness in handling dynamic geometric discontinuities during curb following.

c) Curb following on height-degrading curb segments: An experiment was conducted on an SDU Campus curb segment where the curb height gradually decreased until nearly flush with the road surface [Fig. 5(c), bottom row (right)]. This visually degraded boundary challenged the perception module due to the absence of clear elevation cues. Fig. 16(c) presents the real-time map visualization with corresponding photographs.

The primary objective of this experiment was to assess the system's robustness in tracking curbs with reduced visual salience, especially during transitions from clearly elevated to nearly level curb boundaries. Throughout the trials, the vehicle successfully maintained lateral alignment along the curb path, even as the height contrast between the curb and the road surface diminished. The mean tracking error was measured at 0.0397 m [Fig. 17(c)], confirming the system's stable performance in visually degraded curb conditions. The speed profile shown in Fig. 18(c) indicates consistent velocity regulation up to 1.0 m/s without abrupt decelerations, further validating the system's

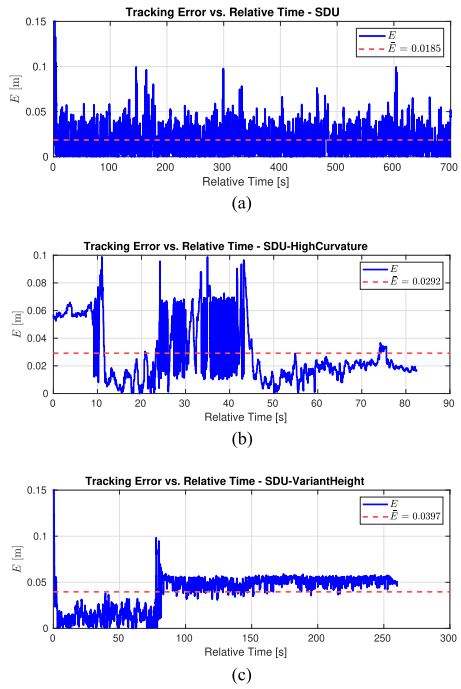


Fig. 17. Real-time tracking error of three different real-world scenarios on the SDU Campus. (a) Tracking error of real-world experiment in the Circular Square of SDU Campus. (b) Tracking error of real-world experiment on high-curvature segments with geometric discontinuities on the SDU Campus. (c) Tracking error of real-world experiment on height-degrading curb segments on the SDU Campus.

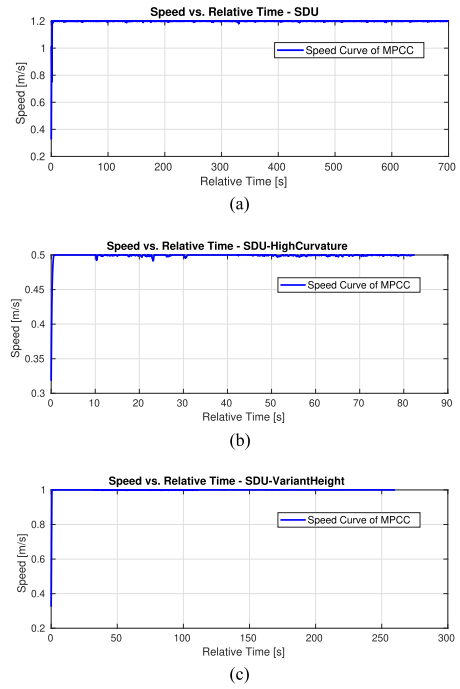


Fig. 18. Vehicle speed profile of real-world experiments on the SDU Campus. (a) Vehicle speed profile of real-world experiment in the Circular Square of SDU Campus. (b) Vehicle speed profile of real-world experiment on high-curvature segments with geometric discontinuities on the SDU Campus. (c) Vehicle speed profile of real-world experiment on height-degrading curb segments on the SDU Campus.

ability to follow curbs under changing elevation contrast scenarios. These results highlight the adaptability of the perception and control modules to challenging real-world curb geometries.

It is worth noting, however, that the proposed curb detection algorithm relies on a measurable height difference relative to the road surface. In scenarios where the curb becomes completely level with the surrounding pavement, eliminating discernible elevation discontinuity, detection performance may degrade or even fail. This limitation stems from the algorithm's dependence on elevation-based geometric features. Addressing such cases may require integrating complementary perception cues, such as semantic segmentation or texture-based edge detection, to ensure reliable curb identification when elevation contrast is absent.

Overall, these experiments demonstrate the proposed system's robustness and adaptability across diverse challenging curb conditions, achieving both precise curb following and reliable real-world deployment.

V. Conclusion

In this article, we have proposed an integrated curb-following system to systematically address the challenges of curb detection and curb-aligned motion generation. A robust and adaptive curb detection algorithm has been introduced, capable of adapting to varying road conditions without relying on predefined parameters. This ensures accurate extraction of curb information, even in complex environments. In addition, an MPCC-based curb-aligned motion generation method has been developed to enable precise curb following with high execution efficiency. The proposed system has been rigorously validated through both Gazebo simulations and real-world experiments, demonstrating its practicality and reliability for autonomous curb-following tasks in urban environments. Its inherent adaptability and efficiency make it a promising solution for applications, such as road sweeping, road inspection, autonomous navigation, and beyond.

For future work, we aim to enhance the robustness of the curb detection method by reducing its reliance on 2.5-D elevation mapping, whose primary limitation lies in the dependence on a single modality of geometric information. Such reliance may lead to degraded performance in scenarios where elevation contrast between the curb and the road surface is minimal or absent. To address this, we plan to integrate semantic perception techniques, such as semantic segmentation or texture-based edge detection, to provide complementary visual cues, thereby improving curb detection in visually degraded or semantically complex environments. Furthermore, we intend to incorporate task-specific optimization objectives into the MPCC-based curb-aligned motion generation method to improve task completion efficiency and broaden its application potential.

References

[1] T. Chen, B. Dai, D. Liu, J. Song, and Z. Liu, "Velodyne-based curb detection up to 50 meters away," in *Proc. IEEE Intell. Veh. Symp.*, 2015, pp. 241–248.

[2] Y. Zhang, J. Wang, X. Wang, and J. M. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3D-LiDAR sensor," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3981–3991, Dec. 2018.

[3] G. Wang, J. Wu, R. He, and B. Tian, "Speed and accuracy tradeoff for lidar data based road boundary detection," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 6, pp. 1210–1220, Jun. 2021.

[4] A. S. Huang and S. Teller, "Lane boundary and curb estimation with lateral uncertainties," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 1729–1734.

[5] M. Kellner, U. Hofmann, M. E. Bouzouraa, and N. Stephan, "Multi-cue, model-based detection and mapping of road curb features using stereo vision," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015, pp. 1221–1228.

[6] M. Cheng, Y. Zhang, Y. Su, J. M. Alvarez, and H. Kong, "Curb detection for road and sidewalk detection," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10330–10342, Nov. 2018.

[7] J. Liang, N. Homayounfar, W.-C. Ma, S. Wang, and R. Urtasun, "Convolutional recurrent network for road boundary extraction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9512–9521.

[8] Y. Jung, S.-W. Seo, and S.-W. Kim, "Curb detection and tracking in low-resolution 3 d point clouds based on optimization framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3893–3908, Sep. 2020.

[9] J. Gao, H. Jie, B. Xu, L. Liu, J. Hu, and W. Liu, "LCDet: Lidar curb detection network with transformer," in *Proc. Int. Joint Conf. Neural Netw.*, 2023, pp. 1–9.

[10] S.-H. Kim, "Tracking control for reliable outdoor navigation using curb detection," in *Recent Advances in Mobile Robotics*. London, U.K.: IntechOpen 2011.

[11] J. Bao, X. Yao, H. Tang, and A. Song, "Detecting untraversable regions for navigating mobile robot on pedestrian lanes," in *Proc. Int. Conf. Intell. Robot. Appl.*, 2019, pp. 661–671.

[12] M. A. Ali and M. Mailah, "Path planning and control of mobile robot in road environments using sensor fusion and active force control," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2176–2195, Mar. 2019.

[13] P. Foehn, A. Romero, and D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight," *Sci. Robot.*, vol. 6, no. 56, 2021, Art. no.eabh1221.

[14] G. Ryou, E. Tal, and S. Karaman, "Multi-fidelity black-box optimization for time-optimal quadrotor maneuvers," *Int. J. Robot. Res.*, vol. 40, no. 1214, pp. 1352–1369, 2021.

[15] A. Rucco, G. Notarstefano, and J. Hauser, "An efficient minimum-time trajectory generation strategy for two-track car vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1505–1519, Jul. 2015.

[16] F. Christ, A. Wischnewski, A. Heilmeier, and B. Lohmann, "Time-optimal trajectory planning for a race car considering variable tyre-road friction coefficients," *Veh. Syst. Dyn.*, vol. 59, no. 4, pp. 588–612, 2021.

[17] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, and W. Burgard, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA, USA: MIT press, 2005.

[18] S.-J. Wu, H.-H. Chiang, J.-W. Perng, C.-J. Chen, B.-F. Wu, and T.-T. Lee, "The heterogeneous systems integration design and implementation for lane keeping on a vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 246–263, Jun. 2008.

[19] A. Amditis et al., "A situation-adaptive lane-keeping support system: Overview of the safelane approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 617–629, Sep. 2010.

[20] O. Törő, T. Becsi, and S. Aradi, "Design of lane keeping algorithm of autonomous vehicle," *Periodica Polytechnica Transp. Eng.*, vol. 44, no. 1, pp. 60–68, 2016.

[21] Z. Chen and X. Huang, "End-to-end learning for lane keeping of self-driving cars," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 1856–1860.

[22] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3019–3026, Oct. 2018.

[23] J. Guerrero, R. Chapuis, R. Aufrère, L. Malaterre, and F. Marmoiton, "Road curb detection using traversable ground segmentation: Application to autonomous shuttle vehicle navigation," in *Proc. 16th Int. Conf. Control, Automat., Robot. Vis.*, 2020, pp. 266–272.

[24] L. M. Romero, J. A. Guerrero, and G. Romero, "Road curb detection: A historical survey," *Sensors*, vol. 21, no. 21, 2021, Art. no. 6952.

[25] F. Ma, P. Hou, Y. Liu, Y. Liu, M. Liu, and J. Ma, "Annotation-free curb detection leveraging altitude difference image," 2024, *arXiv:2409.20171*.

[26] F. Oniga and S. Nedevschi, "Curb detection for driving assistance systems: A cubic spline-based approach," in *Proc. IEEE Intell. Veh. Symp.*, 2011, pp. 945–950.

[27] M. Kellner, M. E. Bouzouraa, and U. Hofmann, "Road curb detection based on different elevation mapping techniques," in *Proc. IEEE Intell. Veh. Symp.*, 2014, pp. 1217–1224.

[28] J. Maye, R. Kaestner, and R. Siegwart, "Curb detection for a pedestrian robot in urban environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 367–373.

[29] X. Mi, B. Yang, Z. Dong, C. Chen, and J. Gu, "Automated 3D road boundary extraction and vectorization using MLS point clouds," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5287–5297, Jun. 2022.

[30] H. Jie et al., "An efficient curb detection and tracking method for intelligent vehicles via a high-resolution 3D-LiDAR," in *Proc. 4th Int. Conf. Inf. Sci., Elect., Automat. Eng.*, vol. 12257, 2022, pp. 310–317.

[31] T. Suleymanov, L. Kunze, and P. Newman, "Online inference and detection of curbs in partially occluded scenes with sparse lidar," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 2693–2700.

[32] Y. Jung, M. Jeon, C. Kim, S.-W. Seo, and S.-W. Kim, "Uncertainty-aware fast curb detection using convolutional networks in point clouds," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 12882–12888.

[33] G. Zhao, F. Ma, W. Qi, Y. Liu, M. Liu, and J. Ma, "CurbNet: Curb detection framework based on LiDAR point cloud segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 6, pp. 8961–8974, Jun. 2025.

[34] Y. Chen et al., "Falcon: Fused attention for lidar-camera curb detection," 2024, *arXiv:2403.16794*.

[35] A. Ošep, T. Meinhardt, F. Ferroni, N. Peri, D. Ramanan, and L. Leal-Taixé, "Better call SAL: Towards learning to segment anything in LiDAR," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 71–90.

[36] S.-H. Kim, C.-W. Roh, S.-C. Kang, and M.-Y. Park, "Outdoor navigation of a mobile robot using differential GPS and curb detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3414–3419.

[37] S. Sivakanthan et al., "Automated curb recognition and negotiation for robotic wheelchairs," *Sensors*, vol. 21, no. 23, 2021, Art. no. 7810.

[38] Y. Wang, Y. Dai, and D. Wang, "Real-time path generation and alignment control for autonomous curb following," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2024, pp. 3175–3181.

[39] R. He, H. Lv, S. Zhang, D. Zhang, and H. Zhang, "Lane following method based on improved DDPG algorithm," *Sensors*, vol. 21, no. 14, 2021, Art. no. 4827.

[40] A. Khanum, C.-Y. Lee, and C.-S. Yang, "Deep-learning-based network for lane following in autonomous vehicles," *Electronics*, vol. 11, no. 19, 2022, Art. no. 3084.

[41] Hako GmbH, Bad Oldesloe, Germany, "Citymaster outdoor cleaning machines." Accessed: Aug. 15, 2025. [Online]. Available: https://www.hako.com

[42] Autowise.ai, Shanghai, China, "Autonomous sweeping vehicle." Accessed: Aug. 15, 2025. [Online]. Available: https://autowise.ai/en/company/

[43] Gaussian Robotics, Singapore, "Autonomous cleaning robots." Accessed: Aug. 15, 2025. [Online]. Available: https://www.gaussianrobotics.com

[44] ENWAY, Berlin, Germany, "Autonomous sweepers." Accessed: Aug. 15, 2025. [Online]. Available: https://www.enway.ai/

[45] E. Prassler, D. Schwammkrug, B. Rohrmoser, and G. Schmidl, "A robotic road sweeper," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia*, vol. 3, 2000, pp. 2364–2369.

[46] J. Jeon et al., "Autonomous robotic street sweeping: Initial attempt for curbside sweeping," in *Proc. IEEE Int. Conf. Consum. Electron.*, 2017, pp. 72–73.

[47] D. Lam, C. Manzie, and M. Good, "Model predictive contouring control," in *Proc. 49th IEEE Conf. Decis. Control*, 2010, pp. 6137–6142.

[48] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. Int. Conf. Knowl. Discov. Data Mining*, vol. 96, no. 34, 1996, pp. 226–231.

[49] G. Welch et al., "An introduction to the Kalman filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, TR 95-041, 1995.

[50] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1: 43 scale RC cars," *Optimal Control Appl. Methods*, vol. 36, no. 5, pp. 628–647, 2015.

[51] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Trans. Robot.*, vol. 38, no. 6, pp. 3340–3356, Dec. 2022.

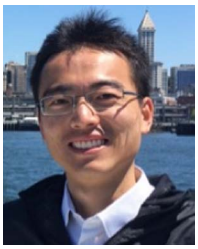
[52] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Math. Program. Computation*, vol. 12, no. 4, pp. 637–672, 2020, doi: 10.1007/s12532-020-00179-2.

[53] T.-M. Nguyen et al., "MCD: Diverse large-scale multi-campus dataset for robot perception," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 22304–22313. [Online]. Available: https://mcdviral.github.io/

[54] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419841532.
[55] F. Oniga and S. Nedevschi, "Polynomial curb detection based on dense stereovision for driving assistance," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst.*, 2010, pp. 1110–1115.
[56] F. Kuhne, W. F. Lages, and J. G. da Silva Jr, "Model predictive control of a mobile robot using linearization," in *Proc. Mechatron. Robot.*, vol. 4, 2004, pp. 525–530.

Jiahao Liang received the B.Eng. degree in vehicle engineering from Tongji University, Shanghai, China, in 2021, and the M.Sc. degree from Nanyang Technological University (NTU), Singapore, in 2022. He is currently working toward the Ph.D. degree with the School of Electrical and Electronic Engineering, NTU.

He is also currently a visiting Ph.D. student with the School of Control Science and Engineering, Shandong University, Jinan, China. His research interests include motion planning, control, and robotic system design.



Yuanzhe Wang (Member, IEEE) received the B.Eng. degree from Southeast University, Nanjing, China, in 2010, the M.Eng. degree from Beihang University, Beijing, China, in 2013, and the Ph.D. degree from Nanyang Technological University (NTU), Singapore, in 2019.

From 2018 to 2024, he held successive positions as Research Associate, Research Fellow, and Research Assistant Professor with the School of Electrical and Electronic Engineering, NTU. He is currently a Professor with the School of Control Science and Engineering, Shandong University, Jinan, China, and also holds a joint professorship with the Shenzhen Loop Area Institute, Shenzhen, China. His research interests include robot navigation, manipulation, multirobot systems, and human–robot interaction.

Dr. Wang serves as an Associate Editor for the *Journal of Field Robotics*, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, and IEEE CONTROL SYSTEMS LETTERS. He has also been an Associate Editor for the IEEE/RSJ International Conference on Intelligent Robots and Systems since 2020.

Guohao Peng received the B.Eng. degree in information science and engineering from Shandong University, Jinan, China, in 2016, and the M.Sc. and Ph.D. degrees in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2017 and 2023, respectively.

He is currently a Research Fellow with the School of Electrical and Electronic eEngineering, Nanyang Technological University, Singapore. His research interests include computer vision, robotics, deep learning, and information fusion.

Zhenyu Wu received the B.Eng. degree of electrical engineering and automation from Wuhan University, Wuhan, China, in 2016, and the M.Sc. and Ph.D. degrees in electrical and electronic engineering from Nanyang Technological University (NTU), Singapore, in 2017 and 2022, respectively.

He is currently a Research Assistant Professor with the Centre for Advanced Robotics Technology Innovation, NTU. His research interests include intelligent perception, localization, and navigation for autonomous robots in complex environments.

Dr. Wu was the recipient of the 2024 IEEE International Conference on Control, Automation, Robotics and Vision Best Paper Award. He served as an Associate Editor for 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Danwei Wang (Life Fellow, IEEE) received the B.E. degree from the South China University of Technology, Guangzhou, China, in 1982, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 1989 and 1984, respectively, all in engineering.

He is currently a Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include robotics, control engineering, and fault diagnosis.

Dr. Wang is also a Fellow of the Academy of Engineering, Singapore. He was the recipient of the Alexander von Humboldt Fellowship in Germany and is a Distinguished Lecturer of the IEEE Robotics and Automation Society.